Functional connectivity inference from time-series of events and application to computer networks

Antoine Messager

A thesis presented for the degree of
Doctor of Philosophy
University of Sussex
July 2019

Contents

Abstract

Today's commercial and Internet Service Provider's networks are large, heterogeneous, fast-evolving and commonly emit millions of events per second. Monitoring, responding to, and predicting incidents is a key challenge for network management operators. This thesis argues that the concept of functional connectivity, widely used in neuroscience and defined in terms of the presence of statistical dependencies between nodes, can unlock informational value about event logs and assist operators in identifying (and even predicting) service outages. However, existing functional connectivity inference methods are not adapted to event data from computer networks. These methods may either require unavailable models of event propagation, be computationally too costly for large networks and/or long recordings, be not adapted to sparse and discrete activity, and/or assume a static network topology. We thus first describe in depth a major commercial network to highlight the challenges faced by network operators and the opportunities offered by thinking in terms of functional connectivities. Next, using a pair of independent Bernoulli processes as reference, we develop a new statistic aimed at measuring coupling strength by quantifying deviation from independence. However, because many statistics will be large by chance, identifying functional edges from the distribution of statistics over every pair of nodes is challenging. Hence, we then develop a method that infers, in specific contexts, the function that associates each statistic to the probability it accounts for the presence of a functional edge. Next, using the previously described statistic, we propose a novel framework to infer a time-varying functional topology from the time-series of emitted events. Applying this paradigm to two major commercial networks, we show that it can reveal a priori unknown groups of devices providing particular services. Finally, we argue that this thesis has implications beyond assisting network monitoring, such as enabling more robust inference of functional connectivity in neuroscience.

Acknowledgment

I would like first to thank the funder, Moogsoft Ltd., and the people of this company without whom this research would have not been possible. All the work describes patented features of its product. Then I would like to thank my supervisors for the work they have done and George Parisis for the help he provided.

Declaration

I hereby declare that this thesis has not been, and will not be, submitted in whole or in part to another University for the award of any other degree. I also declare that this thesis was composed by myself, under the supervision of Dr Luc Berthouze, Pr István Z. Kiss and Dr. Robert Harper and the work contained therein is my own, except where indicated by citations and in the list of publications and author contributions.

Signature: .... (Antoine Messager)

List of publications and author contributions

- **Chapter 2 - Network events in a large commercial network: what can we learn?** Antoine Messager, George Parisis, Robert Harper, Philip Tee, István Z. Kiss and Luc Berthouze, published in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, July 2018, available at `https://ieeexplore.ieee.org/abstract/document/8406289/`.
  - Antoine Messager contributed toward conceiving the overall goals and design of the study, developed and implemented the Julia code for our analyses, produced all the figures, contributed toward writing the first draft, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - George Parisis contributed toward conceiving the overall goals and design of the study, contributed toward writing the first draft, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Robert Harper contributed toward providing access to the dataset, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Philip Tee contributed toward providing access to the dataset, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - István Z. Kiss reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Luc Berthouze contributed toward conceiving the overall goals and design of the study, contributed toward writing the first draft, reviewed the first draft and agreed upon final additions and changes, contributed towards writing the final draft and submitted the article.
- **Chapter 3 - A new method for the robust characterisation of pairwise statistical dependency between point processes** Antoine Messager, Nicos Georgiou and Luc Berthouze, arxiv version available at `https://arxiv.org/pdf/1904.04813.pdf`. Will be submitted soon to *Physical Review E.*

  - Antoine Messager contributed toward conceiving the overall goals and design of the study, contributed towards the derivation of the exact analytical expressions, developed and implemented the Python code for our analyses, produced all the Figures, contributed toward writing the first draft, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Nicos Georgiou contributed towards the derivation of the exact analytical expressions and of the theorems, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Luc Berthouze contributed toward conceiving the overall goals and design of the study, contributed towards the derivation of the exact analytical expressions and of the theorems, reviewed the first draft and agreed upon final additions and changes, contributed towards writing the final draft and submitted the article.

- **Chapter 5 - Inferring Functional Connectivity from Time-series of Events in Large Scale Network Deployments** Antoine Messager, George Parisis, István Z. Kiss, Robert Harper, Philip Tee and Luc Berthouze, submitted to *IEEE Transactions on Network and Service Management*, accepted.
  - Antoine Messager contributed toward conceiving the overall goals and design of the study, developed and implemented the Python code for our analyses, produced all the Figures, wrote the first draft, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - George Parisis reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Robert Harper contributed toward providing access to the datasets, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - Philip Tee contributed toward providing access to the datasets, reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.
  - István Z. Kiss reviewed the first draft and agreed upon final additions and changes, and contributed towards writing the final draft.

- Luc Berthouze contributed toward conceiving the overall goals and design of the study, reviewed the first draft and agreed upon final additions and changes, contributed towards writing the final draft and submitted the article.

Chapter 1

Background and motivations

1.1 Introduction to networks

1.1.1 What are networks?

A brief history

In 1736, Leonhard Euler was in Königsberg, Prussia, (now Kaliningrad, Russia) and set himself an apparently simple problem. The city was set on both sides of the Pregel river and included two large islands connected by 7 different bridges, see Figure 1.1. Leonhard Euler wondered whether it was possible to cross every bridges once and only once.

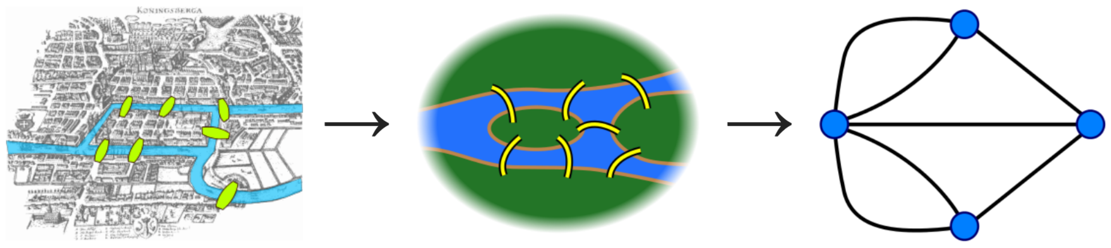


Figure 1.1: Map of Königsberg in Euler's time showing the actual layout of the seven bridges, highlighting the river Pregel and the bridges (left) and progressive abstraction steps toward a graph representation (middle and right). Image reproduced without changes from Wikipedia. Files are all licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license [167, 168, 169].

He realised that whenever one enters a land through a bridge, one leaves the land through another bridge. This obvious statement has an important consequence: every land having an odd number of bridges must either be the beginning or the end of the path.

Yet, every 4 lands of the city of Königsberg are connected to three 3 bridges. Hence, since a path can only have 2 extremities, it proved that no one could cross the 7 bridges once and only once [43]. This property is generalizable: whatever the number of lands and bridges, a path that crosses once and only once every bridges can only exist if there is either 0 or 2 lands being connected to an odd number of bridges. Provided that there exists a path connecting any two lands, this necessary condition was later proved to be sufficient [9]. This generalisation called for a more abstract representation of the problem. It is clear that the size, the shape and the geographical position of the lands and bridges do not matter. This suggests hence that it is possible to remove all the features and to represent the problem in a more abstract way, see a possible evolution of the city representation on Figure 1.1.

The representation of the rightmost part of Figure 1.1 is today known as a graph or a network. Likewise, lands and bridges would today be called nodes (or vertices) and edges. The problem formulated by Euler is hence considered to be the first one within the field of graph theory [112]. Graph theory is the study of graphs which are mathematical structures used to model pairwise relationships between objects. One of the most famous problem within this field is the four colours problem formulated by Francis Guthrie in 1852 and proved true more than a century later: "can any map of regions drawn in the plane be coloured with only 4 different colours such that no two adjacent regions have the same colour?" [22]. From then on, the field of graph theory expended quickly. In 1845 for instance, the physicist Gustav Kirchhoff used a branch of this field, to publish his Kirchhoff's circuit laws for calculating the voltage and current in electric circuits.

Network definition

A graph is an ordered pair $G = (V, E)$ made of:

- V a set of vertices (or nodes)
- $E \subseteq \{(x, y), x \in V, y \in V\}$ a set of edges

A graph can be undirected or directed and weighted or unweighted, see Figure 1.2. When a graph is weighted, a function $w : E \to \mathbb{R}$ is associated to the set of vertices and edges and assigns a real value to every edges of the graph.

Undirected/directed and weighted/unweighted graphs are often represented by their adjacency matrix. It is the $n_v \times n_v$ matrix, where $n_v = |V|$ is the number of vertices,

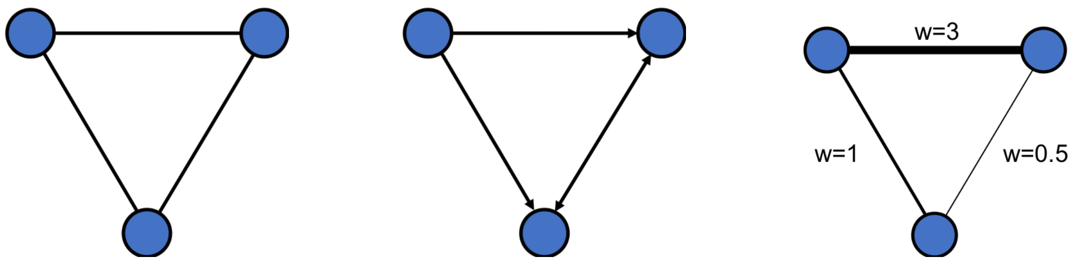
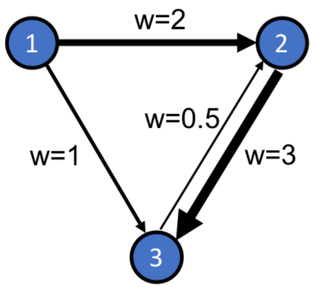


Figure 1.2: Examples of graphs made of 3 vertices and 3 undirected unweighted edges (left), 4 unweighted directed edges (centre) and 3 undirected weighted edges (right).

whose elements indicate whether pair of vertices are linked by an edge, see Figure 1.3. If the graph is undirected, the matrix is symmetric. If the graph is unweighted, the matrix is binary, i.e. only made of 0s and 1s.



	1	2	3
1	0	2	1
2	0	0	3
3	0	0.5	0

Figure 1.3: Example of weighted directed graph and of its adjacency matrix.

Network thinking

The hopes put into network science are high, quoting Albert-László Barabási: "Network thinking is poised to invade all domains of human activity and most fields of human inquiry. It is more than another useful perspective or tool. Networks are by their very nature the fabric of most complex systems, and nodes and links deeply infuse all strategies aimed at approaching our interlocked universe." [6] A few important graph features are worth mentioning to illustrate how network thinking is highlighting common structures across different fields. The most important one is probably the degree of a node. For undirected graph it is equal to the number of incident edges of each node. Note that for directed graph, one has to differentiate the in-degree from the out-degree. The concept of degree and its distribution over every network nodes is key to the analysis of many complex phenomena. In social networks, high degree nodes correspond to the few people that have many friends while low degree nodes correspond to the majority of people who have fewer friends. Similarly, compared to the majority of webpages, there are few very popular websites like Google which links to billions of different webpages. High degree nodes are sometimes referred to as hubs and often have very central roles; they tend to be major conduits for the spread of information across network [161].

The concept of the clustering is another important feature of network analysis. Clustering measures the propensity of each node to have neighbouring nodes attached to other neighbouring nodes of the original one. Tightly connected communities are characterised by high clustering values. Interestingly, many real world networks from social networks, to the world wide web and brains, have been found to exhibit high-clustering, skewed degree distribution and hub structures [7]. We know that such structures have not evolved by accident but for some reason that transcends individual fields [105]. Although it is not yet known why such structures have appeared, this illustrates how network thinking is providing a common languages to the analysis of complex systems across disciplines.

Today, graph theory is widely recognised and is being used in many different fields such as computer science, linguistics, physics, chemistry, meteorology, social sciences, biology... Application of graph theory to these fields have helped answering questions as diverse as: why is the life expectancy of organisms a power function of their size [166]? Why do "fake news" spread so quickly [162]? Why is the Internet very robust to failures in some circumstances and prone to catastrophes in others [28]? What is the weather forecast for tomorrow [156]?

1.1.2 Network across disciplines

Networks are used in a wide variety of domains that are classified in no less than 11 different domains comprising at least 39 different subdomains ranging from biology, to internet, economy and epidemiology [35]. A complete descriptions of every field of applications would be out of the scope of the current work. Instead, we describe below briefly the two most important domains of application relevant to this thesis: computer and brain networks. Note that a more in depth description of the characteristics of computer networks is provided in Chapter 2.

Computer networks

Computer networks are networks of devices connected together via data links that can be physical links, e.g. ethernet cable, or wireless media, e.g. wifi connection. The aim of such network is to share information and to allow users access different services such as the World Wide Web, video and audio streaming, emails, storage servers. The network devices that emit, route and receive the data are called network nodes. They include hosts, e.g. personal computers, phones, servers, as well networking hardware such as networks and switches. These nodes are generally identified by their network addresses, e.g. Internet Protocol (IP) address. However, this IP address is not the only one they possess. When data is exchanged between two hosts, the transmission follows a protocol that specifies how data should be packetised, addressed, transmitted, routed, and received. This protocol divides the network in five abstraction layers, which, according to Transmission Control Protocol and Internet Protocol (TCP/IP) are:

- The physical layer ensures the physical transmission of bits across a physical data link. It defines the bit stream, e.g. electrical impulse, light, radio, through physical link.
- The link layer ensures the transmission of packets between two interfaces on the same link. The interfaces are then identified by a unique physical address.
- The network or IP layer provides routing technologies and uses virtual paths between network nodes. This allows the transfer of data from one network to another one. Nodes are here identified by their IP address.
- The transport layer provides transparent transfer of data between hosts. It ensures complete data transfer.
- The application layer ensures a correct communication between applications of different devices and ultimately between different end-users.

In order to understand the importance of such layers for the description of computer networks, a simplified example of transmission of information between two hosts is useful. Suppose an emitter host wants to emit information, e.g. a mail, to a receiver host. First using protocols specific to the application layer, the emitter automatically transforms the information within the application, e.g. mail application, into transferable data. Then, according to the transport layer protocols the data is divided it into several smaller packets and so to facilitate the transmission of the information. Finally, the emitter relies on network layer protocols to route each packet to the receiver. Each data packet is routed over the Internet from one router to one of its neighbours which are chosen based on their susceptibility to being closer to the receiver. This is done by using routing tables and IP addresses. When all packets are received by the receiver they are re-assembled according to transport layer protocols and then de-transformed, according to protocols belonging to the application layer. Note that we did not mention the link layer not the physical layer. This is because form the view point of the network (link) layer, the link (physical) layer is invisible. When a packet is transmitted between two network layer devices, the packet is transferred via link layer protocols between all the link layer devices, e.g. switches, that are on the path between the two Internet layer devices. There may be many link layer devices between two routers, but from the perspective of the network layer, it is as if those routers are directly connected. Likewise from the perspective of the application layer, devices belonging to the network layer enabling the routing of the packets are invisible.

Depending on the layer that one considers, the network topologies are very different. While the one obtained at the link layer level is close to the physical one and thus quite static, the network level one is a virtual one, omits many physical intermediaries, and is often updated. To add to the complexity of the descriptions, remark that there are devices, e.g. multilayer switches, that can act upon two layers. Hence, when network operators aim at obtaining a mapping of their topology (this is usually done at the network layer level), the observed topologies may be a mixture between link layer and Internet layer devices. Finally, networks observed at the application layer level are also very different from the other two. At this level, the topology is even more virtual since it simply connects applications of different devices that communicate with each other and omits all the intermediaries. Besides, since it grows and shrinks as communications start and stop, it is quickly evolving.

Network of the brain

The brain is an organ that serves as the centre of nervous systems in all invertebrate and most invertebrates species. The building blocks of the brain are specialised cells called neurones. A typical neurone consists of a soma, dendrites and a single axon. The soma is the portion of a neurone that contains its cell nucleus. The axon and dendrites are filaments that grow out of the soma toward other neurones. While dendrites branch profusely and remain a few hundred micrometers away from the soma, axons are unique and can travel for as far as 1 meter within human bodies. At the end of the axon are axon terminals, where neurones typically connect to other neurones' dendrites via synapses, see Figure 1.4. Across species and individuals, neurones typically behave similarly. In the absence of stimulation, neurones maintain a voltage gradient across their membrane. If, for some reason, the neurone's voltage reaches a threshold, it then generates an electric spike that travels rapidly along its axon while reseting the membrane potential to its resting value. When a spike reaches the end of an axon, it typically triggers the liberation of chemical neurotransmitters within each of the axon's synapses. These transmitters are then captured by post-synaptic neurone receptors which leads to an increase (or a decrease) of the post-synaptic neurone membrane potential. If the post-synaptic neurone sees its voltage increase, the pre-synaptic neurone is excitatory. It is inhibitory otherwise.

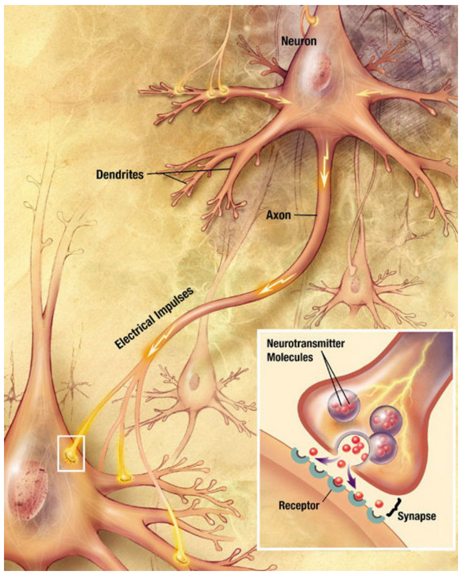


Figure 1.4: Schematic illustration of the connection of one neurone (top right) to another one (bottom left) via its axon and zoom (bottom right) of the axon's synapse. As denoted by the yellow arrow a signal propagating down an axon to the cell body and dendrites of the next cell. Image reproduced without changes from Wikipedia [170]. The file belongs to the public domain.

From the 2000s, researchers started to model the brain as a network where neurones and synapses are modelled by nodes and edges [145]. However brains vary a lot across species. While human brains possess approximately 100 billions neurones each connected on average to ten thousands neurones, *C. elegans*, a 1mm long nematode, possess only 302 neurones [1]. Similarly to the Internet, there exist different levels of mapping. Because of its size no human brain has ever been mapped at the neurone level. Instead, most topologies have been obtained at a coarser level whereby groups of neurones have been used as single nodes. At a mesoscopic scale, they are arranged into networks of columns and mini-columns, while at a macroscopic scale, they are arranged into very large numbers of neurones and neuronal populations forming distinct brain regions [144]. However, at such scales, the definition of edges are not straightforward; they do not account for unique observable physical links. We explore this issue in the following section.

1.1.3 Inferring networks

As exemplified by our brief descriptions of brain and and computer networks, many systems can be described by several network representations. Depending on how one defines nodes, e.g. individual entities or groups of them, and critically how one define edges, network definition changes.

How to infer their connectivity?

The most obvious way to obtain a connectivity would be to list all the observed edges between nodes. However it is often a very complex task: human brains, for instance, possess trillions of synapses. There are tools, such as diffusion tensor imaging (DTI), that aim at reconstructing axons by tracking the diffusion of water, however their spatial precision is too weak to fully reconstruct brain connectivity [144]. Hence many common methods also rely on procedure using data produced by such networks [25]. We have seen that neurones occasionally emit spikes when excited by their neighbours. This generates an electrical activity that can be observed. At a micro-scale, tools like micro-electrodes arrays can be used to measure electric activities of up to thousands of individual neurones over time [27]. At a macro-scale, electroencephalography (EEG) can also be used, although it measures the activity of larger regions. To complement this measurements other techniques exist. They usually rely on proxies measuring electrical activity. For instance, magnetic fields are known to be produced along electrical activity, hence magnetoencephalography (MEG) can be used to measure a magnetic activity from which an electrical activity can be inferred. Likewise, blood flow is also known to be coupled with brain activity. Techniques like positron emission tomography (PET) or functional magnetic resonance imaging (FMRI) can be used to measure blood flow and hence electric activity. From these tools, one generally obtains the evolution over time of activities. They are typically modelled by time-series at particular locations, e.g. at individual neurones level for micro-electrodes arrays. A lot of work has focused on recovering the connections from these observed dynamics and results provided by such approaches have been firmly established [23].

Interestingly, similar dynamics can also be observed on computer networks. As we have seen, computer networks are initially designed to enable data flow. However networking devices are prone to failures. Thus in order to prevent user experience disruption, network operators need to know which devices are facing problems and thus most likely to fail [94]. They hence usually rely on a continuous collection of events from all network devices believed to be important [151]. To do so, a large number of devices are regularly probed and programmed to emit events to monitoring machine when something unusual occurs (we provide in chapter 2 an in-depth description of such events). As a result, network management companies collect a very large number of events providing information about the state of a very large number of devices. Hence, if one only considers events' timing, the obtained dataset is very similar to the one obtained with micro-electrodes arrays. In both cases, one records the evolution of a binary activity, e.g. spikes for the brains and events for computer networks, of every nodes of the network. Within this thesis, we argue that as with neuronal spikes trains, we can recover a connectivity from the events timing recorded by network management companies. We furthermore show that the obtained connectivity is meaningful and that it could not have been inferred without following such an approach. However, before reviewing what are the existing methods that allow one to infer the connectivity from time-series of events, one needs to question what is the nature of the inferred connectivity.

What type of connectivity?

Netwroks' inference from the activity of nodes has been a long standing topic in neuroscience and we typically define 3 types of connectivity [144, 50]:

- The **structural** connectivity refers to a physical network. When considering the brain, the structural connectivity is the anatomical one. It typically refers to a network of synaptic connections linking neurones to each other. Biophysical attributes, such as synaptic strength or effectiveness, may be associated to each edge. The predicted network may thus be directed and weighted (depending on the system studied).

- The **functional** connectivity refers to a network where an edge indicates the presence of statistical dependency between the activity of two nodes. That is to say that there is an edge between two nodes if their activity is dependent on each other. The predicted network is undirected and can be weighted. Note that two nodes may be functionally directly (1-hop) connected and indirectly structurally connected.

- The **effective** connectivity refers to a network where an edge from node A to B accounts for a causal interaction. Activity occurring at node A needs at least sometimes to partially cause activity in B. The predicted network is directed and potentially weighted. Note that effective connectivity is included within the functional one.

The last two connectivities are inferred from the activity of nodes, while the first one ought to be directly observed on the network. It is important to remark that more often than not, it is not possible to recover the structural connectivity from the activity of nodes. Indeed, if a precise, accurate and complete description of events propagation and generation is lacking, the only possible inference relates to the influence of nodes upon each other. By definition, this kind of influence is specific to functional and effective connectivity. In this thesis, we focus on the possibility of inferring a functional connectivity for computer networks. Although it is a well established concept for neuronal networks, it is not for computer networks. We will thus aim throughout this work at demonstrating firstly that it is possible to infer it and secondly that the inferred connectivity is meaningful and characteristic of a functional connectivity. We next review existing methods that infer a functional connectivity from the discrete activity that may be observed on computer networks' nodes.

1.2 Review of functional network inference methods from discrete activity of their nodes

To provide some consistency, we label the n time-series $X_t^i, 1 \leqslant i \leqslant n, 1 \leqslant t \leqslant T$ where $\forall i,\ \forall t,\ X_t^i \in \{0,1\}$. As we will highlight in chapter 2, sparsity is one of the main characteristic of the time-series of events considered within this thesis; we typically consider time-series made of $99,998\%$ of 0s. The ability of every existing method to deal with sparsity is hence discussed in the following. There are two broad categories of methods: parametric and non-parametric [25]. Parametric methods aim at inferring parameters that best a posteriori explain the observed time-series, while non-parametric methods aim at directly measuring statistical dependencies between time-series. Note that many of the methods reviewed here are not uniquely designed for time-series of events, i.e. time-series made of 0s and 1s. As long as the binary nature of the time-series of events does not prevent the use of such methods, we mention them.

1.2.1 Non parametric models

Correlation based analysis

From the 1960s, Gernstein and Clark started to applied Pearson's cross-correlation based methods onto neuronal spiking activities to detect statistical dependencies [55]. From the observed dataset, the cross-correlation between the time-series X^i and X^j at lag τ is estimated via:

$$\rho_\tau^{ij} = \frac{\sum_{t=\tau+1}^{T}(X_{t-\tau}^i - \bar{X}^i)(X_t^j - \bar{X}^j)}{\sqrt{\sum_{t=\tau+1}^{T}((X_{t-\tau}^i - \bar{X}^i)^2}\sqrt{\sum_{t=\tau+1}^{T}(X_t^j - \bar{X}^j)^2}} \tag{1.1}$$

where $\bar{X}^i$ accounts for the average value of X^i. The simplest technique consists in considering only instantaneous interaction, i.e. at lag $\tau = 0$ and declaring the presence of an edge if ρ_0^{ij} is larger than a predefined threshold [179, 13, 158, 40, 79]. This technique suffers however from a few issues. First it requires the interaction to be instantaneous, so if i triggers activity in j with a slight delay, the interaction may not be observed. Second, it is bin dependent: depending on the sampling rate, no events may be co-occurrent and hence no interaction may be detected. Third, the choice of the threshold is problematic: no confidence level is necessarily associated with the correlation (this issue is specifically discussed in section 1.2.3). Finally, it assumes stationarity of the dynamics.

To resolve the first problem, many approaches have proposed to consider the maximal value of the cross-correlation over a few lags within a predefined domain D_τ. However selecting the maximal value induces a bias. Indeed, the more lags are considered the larger the expected value of the maximal cross-correlation is. This issue has been overlooked by many studies, e.g. [174], until Kramer et al. proposed to used extremum theory to overcome it [89]. They proposed to apply the Fisher transform ($z \to \frac{1}{2}ln(\frac{1+z}{1-z})$) to the cross-correlation and to normalise it by dividing it by its standard deviation. They assumed that the set made of the normalised cross-correlation measured at every lag was made of iid $\mathcal{N}(0,1)$ variables. They hence compared the maximal value of this set to its expected maximum, i.e. to maximal value of a set of $|D_\tau|$ iid $\mathcal{N}(0,1)$ variables. Although this corrects for the aforementioned bias, it still requires to make a strong assumption, i.e. normality of the normalised cross-correlation. And there is nothing to make us believe that this applies to time-series of events, especially if these time-series are sparse. The normality assumption indeed only holds if the considered time-series are generated by normal processes [46]. Note that anyway, this bias is likely to have a smaller impact than the one induced by the absence of standard ways to estimate the connectivity from the distribution of the cross-correlations, see section 1.2.3.

Counting based methods

From the 1970s, some approaches have been aiming at taking advantage of the discrete nature of the considered time-series [81]. Instead of considering Pearson's cross-correlation, they proposed to count the number of co-occurrent pair of events and compared this figure to the expected one if the processes had been independent. To do so, they considered Poisson processes and, calling X and Y two binary time-series resulting from two independent Poisson point processes of rates λ_X and λ_Y, they calculated for $\tau = 0$, the expected value and standard deviation of the following random variable:

$$Z_\tau = \sum_{t=1+\tau}^{T} \mathbb{1}(X_t = Y_{t-\tau} = 1) \tag{1.2}$$

This variable measures the number of time-stamps in which one can find events emitted by the two process X and Y. We call this number of time-stamps the number co-occurrent pair of events. They then compared the theoretical number of co-occurrent pair of events to the one observed using the observed rates, e.g. $\lambda^i = \frac{1}{T}\sum_{t=1}^{T} X_t^i$. Significant differences were then typically interpreted as evidence of presence of statistical dependencies. Palm et al. referenced several possible comparison approaches leading to different statistics, the most natural one being a z-score [116]. However, as they pointed out, since the Z random variable was not normally distributed, the interpretation of the comparison was not straightforward. Besides, as for the instantaneous correlation, such approaches were very dependent on the sampling frequency and could only account for instantaneous interaction. The works mentioned here indeed only considered $\tau = 0$.

In order to consider non-instantaneous interactions, some approaches have considered that both the expected values and the standard deviation of the random variable Z_τ were independent of τ and thus equal to their value in $\tau = 0$ [138, 39]. This allowed to select the lag τ where the difference was maximal. However this introduces a bias (see section 1.2.1). Although the correction introduced by Kramer could potentially work [89], no method have been proposed to correct it. A decade later, Grün et al. extended the approach to larger groups of two or more nodes. Their approach was based on counting the number of co-occurrent pair (or larger groups of nodes), labelled "unitary events", and comparing it to their expected value using independent Poisson processes as reference [60]. They proposed a non stationary extension based on sliding windows [61] and a bin independent extension based on multiple shifts [62]. This later extension overcame sampling rate related problems and enabled to detect slightly delayed interactions. However, because of the increased complexity, this method and all its following extensions, only compared the observed number of unitary events to the expected one (and not to the standard deviation of such process), making interpretation hard especially for low rates [127]. There has been no proposition to overcome this problem. Instead, the focus quickly drifted toward the detection of statistical dependencies involving more than two individuals, for which many approaches have been developed [122].

Nevertheless, with the increase of computational power, new approaches to better interpret the statistics have been made possible [96, 63]. For each pair of time-series, these approaches consisted in first generating surrogate data by intelligently temporally shifting one or many events within each time-series to both preserve some temporal characteristics and destroy potential statistical dependencies between time-series. The second step involved using those new time-series to obtain the distribution of the desired statistic in case of independence. The comparison of the statistic to its distribution using surrogate data enabled to obtain a confidence level about the intensity of the interaction. For instance, a promising approach has been developed by Louis et al. They have found a way to destroy statistical dependencies while preserving the distribution of the inter events intervals and the local event rates [97]. The main advantage of these approaches relied on the fact that they provided more robust estimations of confidence levels. However, existing applications remained limited to the detection of instantaneous interactions.

Frequency based methods

An effective alternative to the cross-correlation is its frequency domain counterpart, the coherence. The absolute squared coherence, estimated at frequency f, is defined as:

$$(\gamma^{i,j}(f))^2 = \frac{|S^{i,j}(f)|^2}{S^{i,i}(f)S^{i,i}(f)} \tag{1.3}$$

where $S^{i,i}(f)$ and $S^{j,j}(f)$ are the estimations of the spectral density of X^i and X^j and where $S^{i,j}$ is the estimation of the cross-spectral density at frequency f. Those functions are the Fourier transform of respectively the auto-correlation and cross-correlation, both of which need to be estimated. A common way to estimate the spectral density functions consists in using Welch's periodogram method [165]. This method divides the signal in L potentially overlapping windows and estimates the overall spectral density by averaging the spectral density estimate obtained on each window. This method can be indifferently applied to time-series or point processes and rigorous confidence intervals have been derived in the absence of overlap between windows. More specifically, in case of independence, the absolute squared coherence has a probability $1-\alpha \in [0;1]$ to be larger than $1-(1-\alpha)^{\frac{1}{L-1}}$ [68]. Note that introduction of overlap between windows reduces noise and approximate confidence levels have then been derived [18, 51]. This framework addresses many of the aforementioned issues. It is not bin dependent. It can detect non instantaneous interaction and confidence levels are provided. However, all the methods mentioned so far assume the signals to be stationary. Besides, because of its application within the frequency domain, it tends to provide poorer results as the density of events and as recording duration decrease. In extreme cases where time-series are made of single events, or where only one window is used, the coherence cannot be estimated.

Since coherence has been widely applied within the field of neuroscience and that neurones are known to exhibit non-stationary dynamics [21], alternatives have quickly been found. The first approaches consisted in dividing the signal in stationary segments and computing the coherence on each of them, the "coherogram" is the resulting 3D (window, frequency node 1, frequency node 2) plot [23]. It however required to assume stationarity on each of the segment. To avoid making such an assumption, an efficient alternative involved using wavelet-based transform [132, 92]. Wavelet coherence is particularly suited to quantifying time varying coherence, since it uses a window size inversely proportional to the desired frequency. Note that confidence levels have been derived [59]. More recently, Halliday et al. developed another alternative based on Kalman filtering in the z-domain [67]. However those approaches are still not well suited to sparse and short time-series of events.

Information theory based methods

Another kind of coupling statistics are based on the information theory work of Shannon [137]. They aim at measuring the amount of information that is transferred between two time-series. The most common undirected association measure is called mutual information. Other directed association measures, e.g. transfer entropy [135], Granger causality [20], exist, but are mostly used for effective network estimation and thus we do not explicit them thereafter. For two random variables X and Y taking values within $\mathcal{X}$ and $\mathcal{Y}$, the mutual information is generally calculated as:

$$MI(X;Y) = H(X) + H(Y) - H(X,Y) \tag{1.4}$$

where H accounts for the entropy of the random variable (measured in bits) and is defined as:

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)log_2(p(x)) \tag{1.5}$$

For time-series of events the definition of the random variable space is problematic. Oldest methods grouped many time-stamps together in large bins and considered the distribution of the number of events per bin [17]. However these approaches do not strictly speaking consider time-series of events and are not designed for the kind of datasets we consider. More recent approaches have been aiming at taking advantage of the discrete nature of the events [148]. To do so, they divide the time-series in spike trains made of the k bins and consider the distribution of "words", e.g. 01001 for a spike train made of 2 events over 5 bins. As the bin width vanishes and the recording becomes infinitely long, the estimated mutual information approaches the true value [77]. Theoretically these techniques overcome all the issues mentioned before except non-stationarity. Besides, they are theoretically able to capture the presence of non-linear interactions contrary to the three previous aforementioned type of methods. However, as the number of bins per spike train increases, the possible number of "words" increase exponentially. This quickly makes the estimation of the probability $p(x)$ extremely hard and thus the computation prohibitively expensive.

Covariance selection models

Graphical methods, also known as covariance selection models, aim at finding the precision matrix, which is the inverse of the covariance matrix, via likelihood maximisation. The most common one is the graphical lasso [155]. Noting the observed vectors $X_t = [X_t^1, X_t^2, ..., X_t^n], 1 \leqslant t \leqslant T$, the precision matrix is symmetric positive definite (S_{++}^p) and obtained via:

$$\Theta = \text{argmax}_{\Theta \in S_{++}^p} \log(\det(\Theta)) - \text{tr}(S\Theta) - \lambda ||\Theta||_1 \tag{1.6}$$

where $*$ is the transpose operator, $S = \frac{1}{T-1} \sum_{t=1}^{T} (X_t - \bar{X})^* (X_t - \bar{X})$ is the empirical covariance matrix, $\bar{X} = \frac{1}{T} \sum_{t=1}^{T} X_t$ is the sample mean, $L = \log(\det(\Theta)) - \text{tr}(S\Theta)$ is the log likelihood of the precision matrix (up to a constant and scale) and $\lambda ||\Theta||_1$ is the lasso penalty enforcing sparsity of the connectivity. Note that, for simplicity, after the calculation of the likelihood, L, constants and scaling factors that are independent of Θ have been removed. This is possible since they do not have any influence onto the calculation of the maxima [65]. The introduction of the lasso penalty enables to reduce the number of variable to infer. This makes thus the problem tractable even when the recording duration is short. Compared to pairwise non parametric methods, the great strength of graphical methods relies on its conditional dependency. That is to say that a non-zero value between two nodes indicates a statistical dependency between those two nodes given all the other nodes and possible interaction. However this estimator is only consistent for multivariate gaussian process [178]. Processes of different nature have been used, see for instance a model based on Poisson distribution for genomic data [2] or a model based on binary data [5]. However these approaches all aim at inferring covariance matrices based on instantaneous interactions. As such, they are not well suited for sparse time-series of events where delays are to be expected between events. These methods are thence very dependent of the sampling frequency. Finally, they mostly assume dynamics to be stationary. We remark that some time-varying extension have been developed, see for instance the work of Hallac et. al [65].

Once the precision matrix Θ has been computed, e.g. as in 1.6, the partial correlation matrix can be computed as:

$$PC_{ij} = \frac{\Theta_{ij}}{\sqrt{\Theta_{ii}\Theta_{jj}}} \tag{1.7}$$

Since the partial correlation is calculated from the precision matrix, it is a conditional statistic [37]. This property makes the partial correlation very efficient, enabling a method based on it to leverage the first prize of the First Neural Connectomics Challenge in 2014 [150]. It nevertheless requires however to estimate the covariance matrix, which is challenging for sparse time-series of events.

1.2.2 Parametric methods

Parametric methods aim at reconstructing the network that best enables to generate the observed data. To do so, they generally predefine a generative model that can engender a pre-specified type of dynamics and then find the parameters of this model.

Generative models

Many different models have been developed for neural spike trains data. However out of the 10 different ones reviewed by Doya et al., most try to estimate the spike rates, i.e. the number of events emitted by unit time [37]. This data type is hence not strictly speaking considering the precise timing of events and is thus not adapted to sparse time-series of events. The most common one that uses precise timing is the generalised linear model (GLM) [142, 139]. The spiking process can then be described by:

$$\begin{cases} X_t^i \sim Ber(\rho_t^i) \\ \rho_t^i = \Phi\Big(A_{i0} + \sum_{j=1}^{N}\sum_{k=1}^{K} A_{ij}(k) X_{t-k}^j\Big) \end{cases} \tag{1.8}$$

where X_t^i is taken from a Bernoulli distribution of parameter ρ_t^i, $A_{ij}(k) \in \mathbb{R}$ models the influence node j as on node i at lag k and are the parameters to be found and Φ is the inverse link function, usually a sigmoid or an exponential. This model is able to take into account precise delayed interactions, however this comes at the cost of a large number of parameters to infer.

The second approach that can make use of the precise timing of the spikes is a maximum entropy model [134]. Maximum entropy models enable to select models that respect some predefined constraints and that maximise entropy, i.e. that make the least claim to being informed beyond the predefined constraints. According to Roudi et. al [125], under the constraints of both a known correlation for every pair $< X^i X^j >$ and a known average firing rate $< X^i >$, the model for the distribution of $\{X^i\}_{1 \leqslant i \leqslant n}$ with the largest entropy is:

$$P\Big(\{X^i\}_{1 \leqslant i \leqslant n}\Big) = \frac{1}{Z} exp\Big(\sum_i b_i X^i + \sum_{i<j} J_{ij} X^i X^j\Big) \tag{1.9}$$

where Z is the partition function ensuring that the probabilities sum to 1, and the parameters b_i and J_{ij} are chosen so that the expected values of the X_i variables lead to the observed correlations ($< X^i X^j >$) and firing rates ($< X^i >$). This approach is an extension of the Ising model where the X^i are spins, the b_i are local magnetic fields acting on each spin, and the J_{ij} are the exchange interactions [57]. As such, this model relies on instantaneous correlation and is computationally very expensive, limiting its applicability to a few tens of neurones [175].

Estimation of the model parameters

Standard approaches to estimate the parameters consist in maximising the likelihood of the observed data being generated by the selected model, e.g. as in equation 1.8. One of the advantages of using GLM relies on the fact that its likelihood can be analytically derived [113]. Denoting θ the parameters and D the observed data, the parameters are then often obtained by maximising the following quantity:

$$\mathcal{L}(\theta|D) = log(p(D|\theta)) \tag{1.10}$$

This is usually done via an iterative algorithm such as a gradient descent method. However, when the recording duration is short and the number of events smaller than number of parameters to estimate, maximum likelihood approaches tend to overfit the data [37]. Hence a common way consists in introducing a regularisation term, such as a L1 or L2 penalty which penalises the L1 norm or L2 norm of the solution. The parameters are then obtained via:

$$\hat{\theta} = \text{argmax}_\theta \; p(D|\theta) - \lambda||\theta|| \tag{1.11}$$

The solution is also found iteratively [32]. There are other ways to reduce the number of parameters to find. An appealing procedure introduces a priori knowledge about the parameters distributions, e.g. $p(\theta) \propto exp(-\theta)$ as in [146], and maximises the posterior defined as:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto p(D|\theta)p(\theta) \tag{1.12}$$

This posterior distribution is often not tractable analytically and careful approximations need to be made, see [31] for a review of different approaches. One of the advantages of the Bayesian approach is that it enables to evaluate the reliability of each parameter and enables to perform model selection when several models are compared [49]. However, a lot of parameters need to be inferred to take into account delayed interactions. These approaches are thus slow and difficult to apply to large networks. Besides, because of the large number of parameters to estimate, they typically see their performance decrease with the size of the network [146].

1.2.3 A posteriori verification

Most of the networks observed in nature are sparse; they are far from being fully connected [93]. However most of the aforementioned tools return a coupling value for every pairs of nodes and topology inference from pairwise coupling values is not straightforward. Besides, the accuracy of the inference needs to be tested.

Choice of the threshold

All approaches described in section 1.2.1 (expect covariance based approach) return an undirected weighted matrix in which each weight $w_{i,j}$ accounts for the coupling strength between nodes i and j. However, because of the statistical nature of the weights, naive approaches that, for instance, consider all positive weights, could lead to many false positives and thus potentially to wrong conclusions [52]. In order to infer the topology, most common approaches involve thresholding the matrix, i.e. keeping all edges which are associated to weights larger than a predefined threshold. However there exists no consensus about the way to define such thresholds. Furthermore, the choice of the thresholding method revealed to have a lot of impact onto the predicted network [78].

Some approaches have used arbitrarily defined thresholds, e.g. $\rho > 0.5$ [104]. This is however arbitrary. Others have compared the results obtained for different thresholds and have selected the most appropriate threshold [79, 90]. This is however time consuming and the definition of the appropriateness is arbitrary. Alternative methods have looked at some of the characteristics of the predicted networks, e.g. size of the minimum spanning tree [153], average degree [10], size of the largest connected components [42], edge density [159], and have aimed at preserving it across individuals and/or trials. That is to say that they predefined a characteristic, e.g. average degree $< k >= 5$, and selected the threshold that allowed predicted networks to have an average degree closest to the desired one. However, this requires often unavailable knowledge about the topology of the predicted network and may hence lead to wrong conclusions [52].

More statistically principled methods exist. When confidence levels are known, e.g. as with coherence, it is possible to limit the false positive rate by applying a false discovery rate (FDR) procedure [11]. Given a desired maximal proportion of false positives, this procedure enables to select appropriate thresholds. However this procedure requires accurate estimations of every edges p-value and does not provide information about the number of edges that are incorrectly not predicted, i.e. false negatives. The first point is often problematic because correct analytical confidence levels are usually only calculable for theoretical distributions, e.g. bivariate normal process when considering Pearson's cross-correlation. Hence, with the increase of computing power, surrogate data are increasingly used to overcome this problem. For instance Kramer et al. proposed a bootstrapping method in the frequency domain that provides robust p-values [89]. Note however that robust estimation of the spectrum may be hard to obtain for sparse time-series of events.

Validation of methods

By definition, a functional connectivity is a network of statistical dependencies. As we have seen, there are many approaches that compute such dependencies and no single method is always better than every others [37, 141]. Every approaches are context dependent. Besides, the predictions produced by each method may differ [78]. It should not be surprising that there exists hence no global consensus about how to validate predictions. Nevertheless, synthetic datasets seem to be commonly used [89, 141]. This consist in first generating a random topology, second simulating events using this topology and a model of the dynamics based on field knowledge, and third comparing the predicted topology to the randomly generated one (the better the match, the better the result). We note however the inferred topology is compared to the structural one and as such this introduces a bias that is overlooked. Other approaches generate correlated multivariate time-series and aim at retrieving the coupling coefficients used to generate the time-series [45, 65]. However a bias persists since it can be argued that the coupling coefficients can be regarded as the structural edges weights (2 hops statistical dependencies are indeed not considered).

Validation of methods against real world dataset is even more complex. Indeed by definition, functional connectivities are an illustration of statistical dependencies. However there exists no unique and perfect way to extract such dependencies and there exists hence no ground truth. Hence, methods commonly analyse the output of the inference and use some expert field knowledge to validate it [65, 88]. This is however not always possible since networks are often not known in advance. This remains an important problem.

1.3 Structure of the thesis

Within this thesis, we seek to propose that the concept of functional connectivity can be applied to computer networks and that it can bring information that is both useful and new. However, such paradigm has never been applied to the field of computer networks. Hence, the challenges faced by anyone willing to infer a functional connectivity of a network of computers may remain mysterious. Besides, the potential advantages brought by such a framework may not be immediately obvious. Finally, because most ISPs need to keep their network private for security reasons, there is very little work published about the events emitted by such networks and about the way these are exploited. Hence, within the second chapter of this thesis, we provide an in-depth analysis of the events that network management companies need to operate. Moogsoft Ltd., the funder of this thesis, provided us with one large scale computer network from a major commercial company. We firstly, use this to provide the reader with an as deep as possible description of the topology and of the temporal and textual structure of the events. Secondly, we show that it is possible to use the textual information that comes with the events to get an accurate idea of the function of most devices within the network. Thirdly, we show that there is a significant number of pair of devices that emit events in a synchronised manner. Fourthly, we show that a larger than at random proportion of synchronised devices tend to share the same function. The combination of these four observations opens up the possibility of inferring a functional topology from the time of the events. However such an inference is likely to be challenging and we use this opportunity to highlight such challenges. Finally, we highlight the opportunities brought by such paradigm while monitoring computer networks.

Inferring a functional connectivity from the events of a computer network is quite challenging. There can be up to millions events per second recorded over all devices while many devices may emit only a few events per week [151]. Furthermore, services provided by networks change over time [58]. This thus implies that functional connectivities are time-dependent and that their dynamics cannot be assumed to be stationary. Finally, with the growth of the Internet, computer networks of ISPs are becoming larger and larger. Network management companies typically operate networks made of tens of thousands of devices over several years. As we have seen in the previous section, none of the existing methods are designed to deal with such challenges. Methods may either be computationally too expensive for large networks and/or long recordings, may be designed to only capture instantaneous correlations, may require stationary dynamics and hence static topologies, may not come with confidence levels and/or may not be adapted to sparse time-series of events... In this thesis, we thus attempt at developing a method that overcomes all the aforementioned challenges.

We divide our work in 3 parts, each making up for a chapter of this thesis. Within the first part, we delve into the mathematical aspect of the problem and aim at developing a pair-wise coupling measure that is both quick to compute and that is adapted to the temporal characteristics of the time-series emitted by network devices. In order to infer a functional connectivity from the time-series of events of network devices, the developed method needs first to be computed quickly over ten of thousands of time-series spanning months and thus millions of time-stamps ($3600\text{seconds} \times 24\text{hours} \times 30\text{days} \approx 2.5 \times 10^6\text{seconds}$). Second, it needs to be able to deal with non-instantaneous correlations. Third, it needs to be equally accurate whatever the number of events emitted. Finally it needs to come with a confidence level so that it is interpretable. The review made in the previous section highlighted the fact that there is currently no method able to deal with all these requirements. Because recordings can be extremely long and that time-series of events are usually sparse, considering time-series as a complete structure would be computationally too expensive. Hence it is judicious to only tconsider event timings. This leads to considering the distribution of the delays between pair of events emitted by different nodes. To measure a statistical dependency, i.e. a functional link, we propose to compare the observed distribution to the expected one in case of independence. To compare the distributions, we propose a new statistic. This statistic is equal to the number of pair of events (emitted by two different processes) that are separated by a delay smaller than a given lag. To model independency, we propose to use a pair of independent Bernoulli processes of known rates. Hence, we derive the exact analytical expressions of the expected value and of the standard deviation of this statistic and prove that it is normally distributed for such independent processes. This makes it possible to derive a Z-score that can easily be interpreted as a measure of deviation from independence. This measure is quick to compute, is able to combine into a single figure statistical dependencies up to a lag and comes with a confidence interval. We test its predictive power within the final part of this chapter and demonstrate its ability to deal with all the aforementioned constraints.

However, inferring the functional connectivity from the distribution of the measure over all pair of nodes is not straightforward. As seen in section 1.2.3, standard approaches either arbitrarily fix a threshold or a maximal p-value, or only limit the false discovery rate, or look for predefined structural characteristics within the predicted network. Out of those, the false positive rate limitation is the most statistically principled approach and ought to be the most common. However, this approach assumes a correct knowledge of every p-values and only puts an upper bound on the number of incorrectly detected edge for any threshold. While limiting the false positive rate may be particularly important when dealing with sensitive issues like cancer treatment, it may not be most appropriate for network inference. We argue that it is preferable to consider a variable, e.g. a F1 score, that takes into account both false positive and false negative rates. However, such a framework currently does not exist. Hence, within the fourth chapter, we develop a method that assigns, under specific constraints, to each pairwise measure its probability to account for an existing edge. We then test the predictive power of the method and demonstrate its ability to infer networks under specified conditions.

The approach described so far is able to quickly infer a functional connectivity from sparse time-series of events using non instantaneous correlations. However it assumes that the dynamics are stationary and thus that the functional topology is not changing over time. Computer networks however constantly adapt to varying demands and regularly deploy/un-deploy services. Hence, this stationarity assumption is unlikely to be correct. Thus, within the fifth chapter of this thesis, we make use of the statistics previously mentioned and develop a method that infers a time-varying functional connectivity from time-series of events. To validate our prediction, using the analysis made within the second chapter, we develop a synthetic dataset mimicking as closely as possible the dynamics observed on real world networks. We then thoroughly test our approach and compare it to two state-of-the-art methods that come closest to ours: the first one provides something akin to functional connectivity but not specific to computer networks while the second one that is computer network specific but only deals with effective connectivity. We then apply our method to two large datasets from two major commercial companies. We show that our method displays a larger predictive power than the two state-of-the-art approaches previously mentioned. Next, we provide evidence of the functionality of the inferred network by using knowledge gained within the second chapter. We besides show that the predicted functional topology links devices that tend to share the same function. Finally, we provide evidence of the novelty of our approach by comparing the inferred functional network to the structural one. We also show that the knowledge gained with this approach could not have been guessed otherwise.

In the final chapter of this thesis, we summarise the main results of the research, highlight the limit of our work and propose further directions. We furthermore discuss the applicability of our work for network management companies and the impact it may have. Finally, we argue that, our work, beyond assisting network monitoring, has implications in a variety of fields such as in neuroscience.

Chapter 2

Network events in a large commercial network: what can we learn?

Abstract: *ISP and commercial networks are complex and thus difficult to characterise and manage. Network operators rely on a continuous flow of event log messages to identify and handle service outages. However, there is little published information about such events and how they are typically exploited. In this chapter, we describe in as much detail as possible the event logs and network topology of a major commercial network. Through analysing the network topology, textual information of events and time of events, we highlight opportunities and challenges brought by such data. In particular, we suggest that the development of methods for inferring functional connectivity could unlock more of the informational value of event log messages and assist network management operators.*

2.1 Introduction

Today's commercial and ISP networks are large, complex, heterogeneous and fast-evolving. They usually contain a large amount of servers placed in data centres that are made of commodity, off-the-shelf network equipment to interconnect them and make them accessible to the world. They span a large number of geographical regions (commonly across continents) and provide end-users with access to network and content services and applications. Middleboxes, which implement a diverse set of in-network functionality, are also crucial for the provision of efficient and secure services [121]. Ensuring continuous service availability for such networks is extremely challenging [58]. Despite continuous innovation in the network data and control planes, innovation in the management plane has been slower [53].

Network operators rely on the continuous collection of event data from all network devices (including servers and workstations at the edges of the network) that are deemed to be important; this is most commonly a very large amount of devices. Event data is then collected and analysed either in real-time or off-line, by specialised software.

A key requirement of any such event analysis technology is the ability to identify (or even predict) an outage as quickly as possible before user experience gets disrupted. This, however, is rendered extremely challenging by two features of the data. First, network events are commonly produced at a very high rate, up to 10^6 events per second [151] making the outage identification process both time- and resource-intensive [88]. Second, the vast majority of these events are just noise and only a few of them correlate to 'actionable events'.

A lot of research has been done on algorithms performing Root Cause Analysis (RCA) [94]; i.e. identifying network events that escalate to actual network and service outages. Steinder et al. identify three main approaches to extracting the causal sources of events. *Rule-based analysis* is commonly used [119]. In this approach, the network operator predefines a set of rules to identify the causal sources of events and exclude uninteresting devices. Rules need to be updated when there are changes in the network. Knowledge from the network topology and temporal correlations between events is exploited to make those approaches more accurate and flexible [3]. In *model traversing techniques* one explores progressively the neighbours of each entity emitting an event to identify its source [83] using a formal representation of the network structure. Finally, *graph-theoretic approaches* have also been proposed. They require a priori knowledge of how failure of one device affects other devices in order to build a causality or dependency graph. Those graphs are then used to return a number of fault hypotheses that best explain the observations. This has been shown to be an NP-hard problem and several heuristics have been developed to reduce the complexity of the problem [19, 84, 87]. Most of those approaches require a priori knowledge about the network and are therefore static. They are ill-suited to networks with multiple layers of logical connectivity and could be greatly improved by the use of temporal correlation between events [72]. They scale poorly with the number of events to be processed, therefore approaches to eliminate insignificant events have been proposed, by manually setting filters that exclude specific devices or types of events from processing by an RCA algorithm. Such static approaches are problematic in dynamic and fast-evolving networks like the ones of modern service and content providers. Recently, dynamic approaches for filtering network devices based on the notion of graph vertex entropy [152, 151] and supervised machine learning [70] have been proposed.

Despite this literature on efficient network event processing, there is actually little published information on the characteristics of network events from both in-network and end-host devices in large, modern, commercial networks. Heterogeneity in the functionality, and therefore the type of network events produced, makes RCA even more challenging. In this chapter, we present a study of network events collected from a large commercial network over a period of five months. Through attempting to reconcile information gathered from both event-related data and the underlying network topology, we aim to complement the community's knowledge about this type of data as well as illustrate some of the challenges faced by network management operators with the view to stimulate new research in that area. In particular, we will argue that network management operators may benefit from methods that can use event logs to infer *functional connectivity*, i.e., a logical topology involved in the provision of a particular service, irrespective of the underlying physical topology. For example, a cluster of load balancers, application and database servers would comprise a meshed functional connectivity, where all nodes are logically connected to one another. We use this dataset to identify requirements and challenges for such methods.

2.2 Dataset Description

This chapter is based on a dataset of network events and the underlying network topology, from a large commercial network[1]. The network is comprised of a core of meshed backbone routers and a distribution layer of switches. The network is primarily used to support the commercial operations of a Fortune 500 technology company, including accounting, human resources, research and development, communications and telephony and sales support activities. There are points of access to this network at almost every major city in the United States and multiple cities in most European and Asian countries. Network events span a duration of five months (2/5/2015 to 25/11/2015).

Information about devices come from two sources: the network topology and a list of event log messages. The given topology is a list of links between different IP addresses. It was recorded at the end of November 2015 and only provides information about the connectivity at that time. Event log messages are collected from various sources in the network (end-hosts, switches, routers and middleboxes) and refer to functionality at various network layers; e.g. application-layer notifications, such as runtime exceptions, as well as

[1]The dataset is currently not publicly available due to its commercially sensitive nature.

network and link layer, such as routing protocol errors and links going up or down. Each event is associated with a textual type, a textual description, a time of emission and an IP address. Hence, each event provides a temporal proof of the existence of its associated device. Events do not however provide information about the connectivity. In the following, we consider that devices are identified by their IP address, irrespectively of the IP address source (event or topology). We analyse the match between different sources in section 2.3. Note that the nature of each device can only be indirectly guessed from the type of its associated events. In section 2.4.1, we explain how we managed to identify the nature of most devices and we provide an analysis of their distribution.

Emitted events that are recognised as pertaining to the same network issue, warning or notification are then grouped into a so-called network *alert*. An alert is a Moogsoft construct[2] defined as "a de-duplicated event or an instance of new data". The grouping of events into alerts is done automatically. Theoretically, two events belong to the same alert if they are emitted by the same device for the same reason. This reason is extracted from the textual description associated to each event. Furthermore, two events emitted by the same device for the same reason should be incorporated into the same alert. To summarize, each alert contains a human-readable type and description, the number of de-duplicated events contained in the alert and timestamps for when the first and last events for the alert were collected. Our dataset exclusively contains alerts. An alert is the user-presentable notion of a potential problem and is therefore handled as a ticket that needs to be 'actioned' by an operator.

2.3 Network Topology

Network management providers may not always be in a position to maintain an accurate view of the network topology of their clients. Indeed, although a client's devices and software (along with in-house management systems) are configured to emit events to the provider's server(s), changes in the underlying topology are not necessarily visible to the network management provider. This is typically because accurate knowledge of the network topology is not necessary to run the operations' monitoring software. Additionally, it may also be challenging to acquire the network topology, especially when changes in it are frequent. Finally, providing access to such information presents significant security risks. However, we will argue in Section 2.6 that it could be automatically inferred from the events emitted by network devices.

[2]`https://docs.moogsoft.com/display/060000/Glossary`

The topology we were provided with was obtained by taking a dump of the customer's operations database. This database is fed by the change management and connectivity discovery systems. It is automatically updated when network links are provisioned/de-provisioned or equipment is configured. The network obtained from the list of paired IP address, consists of 73,677 devices and 117,846 physical connections among these devices. The underlying graph is not fully connected; its largest connected component (referred to as giant component thereafter) consists of 30,229 devices, which is less than half of the total number of devices. Figure 2.1 shows the degree distribution of the giant component (left panel) as well as the size distribution of the connected components (right panel). In the latter, we observe a very large number of very small components ($\sim 15,000$ components of size less than 10 devices each). This is a striking observation which, at first sight, could suggest a seriously flawed topology collection and recording process. However, in Section 2.4.1, we will show how an analysis of the recorded events enables the formulation of hypotheses for such a large number of disconnected components. Importantly, an analysis of the giant component alone reveals key network metrics that are consistent with those typically found in large communication networks [44], namely: the dataset meets Clauset et al.'s "super weak" definition of scale-freeness [34] (although all p-values were equal to 0, the fit between the data and a power-law distribution was visually much better than the one between the data and a log normal, a Weibull or an exponential distribution), it is dissassortative ($r = -0.22$) and has almost-zero clustering coefficient ($c = 0.00011$).

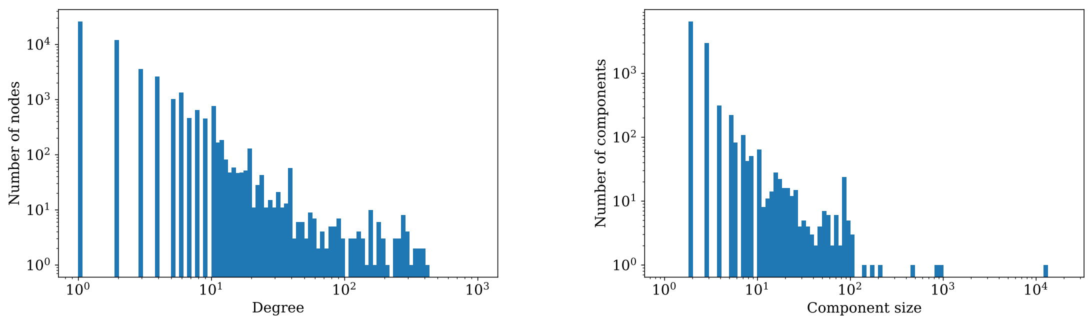


Figure 2.1: Network topology: (left) node degree distribution for the largest connected component and (right) component size distribution.

We conclude this Section with the following two observations: (a) a large fraction of nodes ($\sim 67\%$) in the recorded topology do not have any events associated with them; (b) a large number of events are associated with nodes that are not listed in the recorded topology. With respect to the former observation, we speculate that these are ‘uninteresting’ devices (probably workstations at the edges of the network), which are not configured to communicate with the event processing server but have been recorded in the topology nevertheless. On the other hand, the latter observation highlights the incomplete nature of the topology record, with a part of the network, including a large number of edge devices, not recorded despite its devices being configured to emit events to the server. However, this could also be because nodes that emitted events were no longer in the network (or their identifier changed) when the network topology was recorded. Note that events from these devices are fully usable when it comes to identifying faults with network components or applications, highlighting the fact that partial only knowledge of the topology does not hinder the operation of the network management system. In Section 2.6 we will suggest that such events may provide the substrate to infer missing nodes as well as uncover information about changes in the network topology over time.

2.4 Network Events and Alerts

Among the 53,604 network devices that emit events, 23,919 are part of the recorded network topology ($\sim$ 32% of the total number of nodes in the recorded network topology) and 13,025 are part of the giant component ($\sim$ 42% of the total number of nodes in the giant component). As mentioned in the previous section (and indicated by the percentages above), a large number of nodes in the recorded topology do not emit any events. We considered 21,223,756 events grouped in 473,580 alerts. An analysis of the distribution of number of emitted events per node, shown in Figure 2.2, reveals that most nodes only emit a few events ($\sim$2 events per day). However, there is heterogeneity in behaviour and a few devices in the network emit a large number of events. The average number of emitted events per node indicates that in this dataset there is sparsity of information at node level. This sparsity is not particularly consistent with published numbers but is specific to the particular network presented here. More importantly, it has computational and mathematical implications which we will discuss in Sections 2.5 and 2.6.

2.4.1 Event Types

Every emitted event is accompanied by a type field which usually contains a single word. This field is mapped to specific types of events at deployment time. Such mapping can be manual, e.g. for events received from third-party event management/monitoring systems, or automatic, i.e., involving text processing to extract meaningful types from raw log

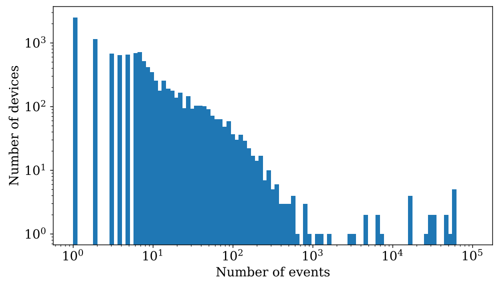


Figure 2.2: Distribution of the number of emitted events per node

messages. Out of a total of 41 distinct types[3], 17 are found to account for over 98% of all events in the record. These are shown in Figure 2.3(left), with type "other" aggregating the remaining 24. For each type, we plot the number of emitted events, recorded alerts and the number of network devices involved in those.

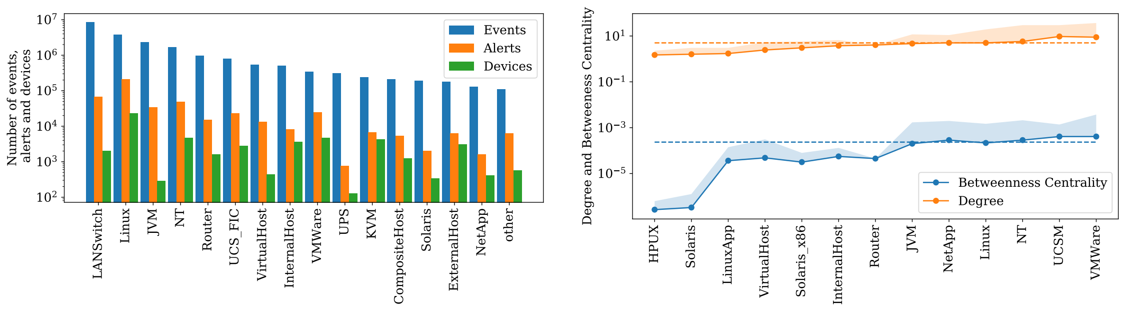


Figure 2.3: (Left) Number of devices (red), recorded alerts (blue) and emitted events (orange) for each event type. (Right) Average degree and betweenness centrality (and their upper standard deviation) of nodes of the giant component who emitted events, organised by event type. For comparison purposes, the dashed lines show the mean degree and betweenness centrality for the full giant component (i.e., including devices that did not emit any event). Lower standard deviations are not shown due to log-scale plotting.

The types of emitted events appear to fall under two categories; those that refer to in-network functionality ('LANSwitch', 'Router') and those that refer to end-host/server functionality (all other types). The former account for almost half of all events (41%) whilst the three most prevalent types in the latter category ('JVM', 'NT' and 'Linux') account for 36% of the events.

Devices emitting 'LANSwitch' or 'Router' events account for only 6.6% of the total number of devices. This is consistent with the assumption that those devices are in-

[3]Roughly 0.5% of the emitted events (1% of the recorded alerts) have an unspecified type field and were therefore excluded from further analysis.

network devices (e.g., routers and switches) as in such a network we can expect the number of devices at the edges of the network to be significantly larger than that of in-network devices. Using the provided network topology, and subject to the caveat of nodes emitting events not being recorded, we found that there is very little overlap between sets of nodes involved in the emission of events of a particular type. Specifically, only 0.5% of all devices emit events of more than one type. In the main, devices seem to be emitting events that pertain to their functionality in the network. For example, we identified that events of type JVM are emitted by web back-end systems when failing to connect to other back-end servers (e.g. database servers). It is possible, of course, that the reason why devices only emit events of a single type can be attributed to choices made by the network management operator. In addition to having a potential impact on the event rate (i.e., this deployment may be less verbose than others), this consideration also has implications for the notion of functional connectivity as we will discuss in Section 2.6.

Devices within the giant component emit events of 13 (out of all 41) types, namely: 'Linux' (8074) - 'VMWare' (2833) - 'NT' (862) - 'VirtualHost' (372) - 'JVM' (289) - 'NetApp' (228) - 'UCSM' (108) - 'Solaris' (15) - 'InternalHost' (12) - 'LinuxApp' (11) - 'HPUX' (2) - 'Solaris_x86' (2) - 'Router' (1). A striking observation is that none of these types pertain to in-network functionality (with the exception of the single event of type 'Router'). This observation is counter-intuitive at first. However, juxtaposed with our previous observation in Section 2.3 that the recorded topology consisted of a large number of small disconnected components, it begs the question of whether some of the devices in the disconnected components may be duplicate entries of devices that are in the giant component. This is indeed plausible given that (a) routers (and their connectivity with other network devices) are typically known by, and recorded with, multiple IP addresses (and respective canonical names) that correspond to interfaces to different IP subnets they interconnect; (b) network switches are also known by, and recorded with, multiple IP addresses for administrative and monitoring purposes; and (c) it is common to use different IP addresses for network management functionality, i.e., for running SNMP and communicating with custom network management software. In Section 2.6, we will appeal to the network science notion of *multiplex networks* and argue that a suitable functional connectivity inference method should make it possible to match these devices.

Within the giant component itself, we find that devices that emit alerts show great variation in their degree and betweenness centrality (see right panel of Figure 2.3), suggesting that devices associated with a type are not randomly distributed within the network. For instance, devices that emit ‘VMWare’-type events tend to be centrally located and more connected than devices emitting events of other types. This yet again brings to fore the notion of functional connectivity.

2.4.2 Alert Descriptions

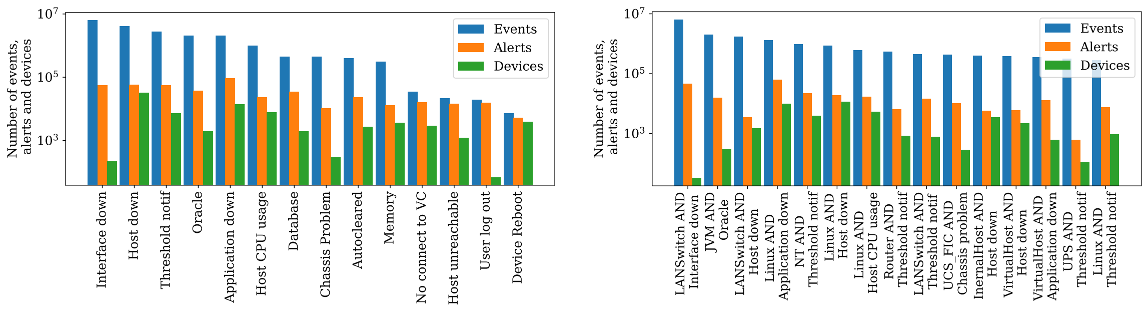


Figure 2.4: Number of devices (red), alerts (blue) and events (orange) for each class of alert (left) and for each of the 15 most populated type-class pairings (right).

The description field of each event contains textual information produced by the source of the event (e.g., a specific Java exception that is pushed to the management system through the *stderr* stream). The vocabulary in the descriptions is very context-specific, containing specialised words and technical jargon (e.g. “host or interface down”, “Database Instance Session exceeding temp threshold”). We used a custom iterative filtering approach to produce a minimal classification of the descriptions. At each iteration, we visually inspected a randomly selected set of unclassified alerts. From their description, we constructed a meaningful pattern of words (within the specific context) as basis for a new class. We automatically searched for all remaining alerts whose description matched the pattern and assigned them to the newly created class. The process was continued until the chosen pattern could not classify more than 1% of the recorded alerts. This iterative process led us to identify 14 classes that account for 94% of the recorded alerts (or 97% of the emitted events). Ninety six percent of all devices are involved in at least one of these classes, which are as follows:

1. Interface:= [name] down (e.g. interface:=eth1 down)
2. Host down: [canonical name]
3. Threshold Notification (e.g. [cpu10]=< 90.00)
4. Application down: [name] (e.g. application down: NTP)

5. Host CPU usage (e.g. Virtual Machine/Host CPU usage)
6. Server Inaccessible (along with chassis and blade information)
7. Auto cleared (e.g. AutoCleared due to inactivity timeout: null)
8. Memory usage problem
9. Database problem
10. Oracle related problem
11. Host not connecting to VC (e.g. ESX host [name] may not be connecting to VC [name])
12. Host unreachable (e.g. [name] Agent is unable to communicate with the OMS (agent is unreachable))
13. User logging out (e.g. User [name] logged out [time])
14. Device rebooting (e.g. [name] rebooted [time])

As shown by the left panel of Figure 2.4, classes 'Interface Down' and 'Host Down' account for 24% of all recorded alerts (50% of all emitted events). Furthermore, these two classes, along with classes 'Threshold Notification', 'Oracle Problem' and 'Application Down', which are notifications or describe application layer events, account for 38% of the recorded alerts (87% of the emitted events). From a number of devices perspective, the 'Host Down' class is the most common, with 59% of all devices emitting events of this type. On the other hand, there are only 228 devices that emit events of class 'Interface Down' class.

An analysis of all type-class pairings revealed that just 15 pairings (right panel of Figure 2.4) account for 53% of the total number of recorded alerts (80% of the total number of events). Strikingly, one of those (LANSwitch and Interface Down) account for 10% of all alerts (33% of all events) despite only involving 33 devices, none of which are located in the giant component (we refer the reader to Section 2.4.1 for potential reasons why devices that emit events relevant to in-network functionality are not part of the giant component). Events of this specific pair indicate links that go down, as identified by neighbouring switches, and are unsurprisingly very common, given the large size of the network, including end-devices.

2.5 Analysis of Temporal Correlations

Since the presence of temporal correlations between emitted event times can provide insights into possible functional relationships between the devices that emit them [94], we used standard correlation techniques (see [99], for example) to gain further insights into the dataset. Namely, we assessed the presence of a functional relationship between two devices in terms of whether the Pearson's correlation coefficient between the time series of their respective recorded alert timestamps significantly differed from that expected under the null hypothesis that they were independent. Formally, this process involves calculating the Pearson's correlation coefficient and applying the Fisher's z-transformation. A pair of time series of recorded alerts was considered to be significantly correlated if their z value was greater than one standard deviation (85% confidence level) [82].

In this dataset only recorded alerts were provided with timestamps and therefore our analysis relied on alerts rather than events. Further, since most network devices are linked to a single type of recorded alerts, we associated each device to its most frequent alert type. Note that due to the very low node-level alert rate reported earlier, binning was needed before analysis. An arbitrary bin size of one hour was used.

Since little could be expected from a pairwise analysis due to the extreme sparsity of the data, we only report summary statistics based on a group-level analysis. In what follows, we will refer to as *within-type pair* a pair of devices that have the same most frequent type of recorded alerts, and *between-type pair*, a pair of devices whose most frequent types of recorded alerts are distinct. For each of the alert types, we calculated the proportion of within-type pairs who showed significant correlations in the time series of their respective recorded alert timestamps and compared it to the proportions of between-type pairs that showed significant correlations. Statistical differences between within-pair proportion and between-pair proportions were assessed using a simple boxplot-based outlier detection method [157].

This analysis revealed that for 18 of the 41 types, within-type pairs, i.e., pairs of devices with recorded alerts of the same type, were more likely to have correlated alert times than between-type pairs, i.e., pairs of devices with recorded alerts of a different type. On the one hand, this increased likelihood is not particularly surprising (especially given that in this dataset devices tend to only emit events of a single type) and provides support to the notion that temporal correlations could enable the identification of devices belonging to the same functional topology. On the other hand, the analysis also strongly suggests that these correlations are not trivial. First, not all event types feature significant correlations in the

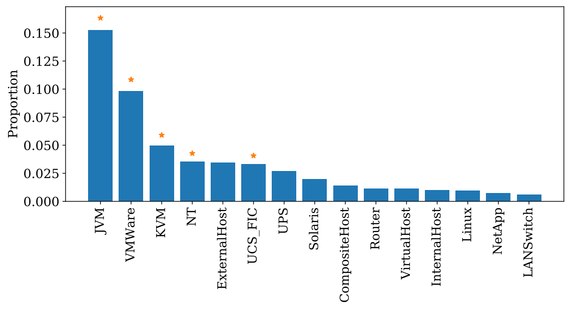


Figure 2.5: Proportion of pairs of devices whose alert types are identical and whose alert times are significantly correlated, for each of the 15 most common types. An asterisk denotes the fact that this proportion is significantly higher than expected when considering all possible pairs of devices, irrespective of type.

alert times of their devices. For example, types 'Router' and 'LANSwitch' (which account for 41% of all events) do not show significant correlations. Further, the 18 types that do show significant correlations only account for 29% of the alerts (25% of the events). Second, as shown by Figure 2.5, in those types that show significant correlations (see asterisk), less than 20% of the pairs show significant correlations. This therefore suggests that correlations are not trivial but may be localised spatially and temporally (see Section 2.6).

2.6 Discussion

Our analysis of a major commercial network has elicited a number of insights which we believe highlight the need for methods able to infer functional topologies within a network deployment based on the temporal characteristics of events emitted by its network devices.

Although the total number of events to be processed by a network management server can be really large, device-level event rates can be very low, which makes device-based analysis very challenging. The reasons for such sparsity are multiple and include whether the device exists at all times during the record, whether or not it is monitored for specific types of events, and indeed whether it is affected by such types of event. In this deployment, network devices tended to emit events of a single type. Although this increased sparsity, it also helped with identifying the role of the devices in the network (e.g., application or database servers, routers or switches) which can be crucial to building a reliable view of the underlying physical topology.

Getting access to reliable information about the physical topology is not straightforward. Yet, such knowledge would be beneficial for a network management provider that would otherwise have to rely solely on incoming events to identify service outages in an unknown (or partially known) network topology. The presence of disconnected components presents a challenge for a traditional graph-based analysis. An alternative approach is to think of the network as a *multiplex network*, i.e., a network where the same set of nodes are linked by different types of interaction. In this approach, each disconnected component may provide a graph-representation of a given layer of interaction. What is needed then is a means to match the nodes across layers of interaction.

Our analysis of time series of recorded alerts in section 2.5 hints at a possible approach to elicit layers of interaction, by showing that correlations exist for alerts of the same type, recorded for different network devices. Although a much more robust framework is needed (e.g., to deal with sparsity and time-varying nature of the signals), the existence of correlations opens up the possibility of inferring common functionality of different devices in the network, i.e., functional topologies. Such functional topologies may be application specific (e.g. a web application development that consists of application and database servers), refer to in-network functionality (e.g. an Open Shortest Path First area), or cut across layers (e.g. when switches report failed links to a database server while application servers try to connect to the same database server). This would provide an extremely powerful framework for network management providers to identify (or even predict) service outages, since functional topologies can be thought of as defining spheres of influence for network devices whereby the operational status or (mis)behaviour of one or more devices would influence the operational status or behaviour of other devices within its sphere of influence.

Chapter 3

A new method for the robust characterisation of pairwise statistical dependency between point processes

NB: The section 3.6 is not part of the paper submitted to PRE.

Abstract: *The robust detection of statistical dependencies between the components of a complex system is a key step in gaining a network-based understanding of the system. Because of their simplicity and low computation cost, pairwise statistics are commonly used in a variety of fields. Those approaches, however, typically suffer from one or more limitations such as lack of confidence intervals requiring reliance on surrogate data, sensitivity to binning, sparsity of the signals, or short duration of the records. In this chapter we develop a method for assessing pairwise dependencies in point processes that overcomes these challenges. Given two independent Bernoulli sequences X and Y each emitting events with probability p_x and p_y and defined over a fixed time period $[\![1;T]\!]$, we derive exact analytical expressions for the expected value and standard deviation of the number of pairs events X_i, Y_j separated by a delay of less than τ. We prove that this statistic is normally distributed in the limit of large T, which enables the definition of a Z-score characterising the likelihood of the observed number of coincident events happening by chance. We numerically confirm the analytical results and show that the property of normality is robust in a wide range of experimental conditions. We then experimentally demonstrate the predictive* *power of the method using a noisy version of the common shock model. Our results show that our approach has excellent behaviour even in scenarios with low event density and/or when the recordings are short.*

3.1 Introduction

Network-based modelling, whereby nodes denote components of the system and edges denote interactions between those components, has become a paradigm of choice for describing, understanding and controlling complex systems [110]. It is widely used in many fields including neuroscience [145], mathematical epidemiology [111] and social sciences [91] to name just a few. However, in many cases, the actual connectivity between individuals may not be directly available or may be incomplete. A substantial body of work has therefore focused on inferring links using the activity of the nodes (e.g. neural spikes, computer events, tweets, gene expression levels) and coupling measures [15, 122]. In this chapter, we focus on the problem of inferring *functional connectivity*, i.e., the graph of pairwise statistical dependences between components of the system as opposed to *causal connectivity* where such dependences are assumed to be causal [129]. Further, we restrict ourselves to systems involving point processes. Although the discrete nature of such processes provides greater analytical tractability, reliable inference of functional coupling between components remains a challenging problem: the exact nature of the point processes may not be known [131]; non linearity, non stationarity and noise are often to be expected [107, 109, 118]; and finally, recordings may be short and repeated measurement unavailable [131].

A common way to quantify functional interaction between two point processes consists in computing Pearson's cross-correlation and selecting its maximal value [89, 180]. Theoretically, after a hyperbolic transformation, the resulting value can then be compared to the theoretical value of the standard deviation and an interpretable Z-score may be extracted [47]. However, there are a number of issues with this approach: it assumes stationarity[1], it is bin dependent, it assumes that recordings are infinitely long, the Z-score is only approximately normal if the processes are bivariate normally distributed (hence making its interpretation somewhat hazardous) and the selection of the maximal value limits the ability to characterise different patterns of interaction, especially if a large range of lags

[1]The issue of stationarity is common to most methods (including our own) in their basic form and is typically addressed through windowing and/or more sophisticated filtering methods.

is considered. Hence, in neuroscience and genetics, in particular, it is still very common to simply threshold the cross-correlation, where the threshold is arbitrarily chosen [26, 41] or based on some expected or desired [10, 73, 163] characteristics of the resulting network. Whilst such an approach removes the assumption of bivariate normal distribution, it is somewhat unsatisfactory because in general the characteristics of the network being inferred are not known. More recent studies rely on comparing the cross-correlation to that of surrogate data. For example, Smith et al. [141] create null data consisting of testing time-series from different subjects, i.e., without causal connections between them. This enables the choice of a threshold and interpretation of the output. However, this does not solve the other aforementioned issues and also introduces the computational cost of generating the surrogate data.

A powerful alternative to the use of cross-correlation to infer functional connectivity is the frequency-domain concept of coherence (along with its somewhat less intuitive time-domain counterpart, the cumulant), applicable to both time series and point processes (or both) and for which rigorous confidence intervals have been derived [69]. Whilst this framework addresses many of the aforementioned issues (and we note recent work extending it to characterising directionality of interaction [66]), its reliance on long recordings can make analysis problematic for short recordings or when the data are sparse [80]. The issue of short and/or sparse data was addressed in much prior work when some sought to count the expected number of coincidental pair of events and derive its expected standard deviation [81]. This allowed the derivation of an interpretable Z-score and was successfully used to separate purely stimulus-induced correlations from intrinsic interneuronal correlation [138, 39, 4]. Similar results were independently found by Palm et al. who thoroughly analysed them [116]. Still, this approach did not address other issues such as bin dependency or the choice of a specific lag.

In recent years, the neuroscience community (in particular) has sought to move away from pairwise statistics and reveal higher order structures [122, 23]. And whilst some of those newer approaches do meet the aforementionned challenges and efficiently recover functional networks (see [33] for example), application to large systems can render them slow and hence not suitable to large networks such as those found in commercial computer networks [103] or social media networks. In addition, these Bayesian approaches typically see their performance decrease with the size of the network due to the increase in the number of parameters to be fitted [147].

In this chapter, we derive a pairwise statistic of interaction between point processes that is computationally inexpensive and that overcomes all aforementioned problems: it is bin independent, it measures interactions at all lags, it does not require that the recording be infinitely long, it comes with confidence intervals. In Section 3.2 we first derive exact analytical expressions for the expected value and standard deviation of the statistic. We then prove that this statistic is normally distributed. In Section 3.3.1, we demonstrate the excellent agreement between analytical and experimental results. In Section 3.3.2, we experimentally confirm that the statistic converges to a normal distribution (as proven previously) and therefore enables the construction of a Z-score. We describe the role of rates and lag in the time needed for convergence. In Section 3.3.3, we use the delayed common shock model to demonstrate the effectiveness of our statistic in characterising both instantaneous and delayed interaction.

3.2 Methods

3.2.1 Graphical construction of the theoretical model

For any integer time horizon T we construct two independent Bernoulli sequences, $\widetilde{\mathbf{X}}_T$ and $\widetilde{\mathbf{Y}}_T$ up to T. To be precise, $\widetilde{\mathbf{X}}_T = (X_1, \ldots, X_T)$ and $\widetilde{\mathbf{Y}}_T = (Y_1, \ldots, Y_T)$. Each coordinate X_i, Y_j are i.i.d. Bernoulli(p_X) and Bernoulli(p_Y) respectively, and each realization is a sequence of 1's and 0's. Parameters p_X (resp. p_Y) is the probability of seeing a 1 in the $\widetilde{\mathbf{X}}_T$ (resp. $\widetilde{\mathbf{Y}}_T$) sequence. On average, by the strong law of large numbers, the expected number of 1's in the $\widetilde{\mathbf{X}}_T$ sequence and the $\widetilde{\mathbf{Y}}_T$ sequence are respectively equal to $p_X T$ and $p_Y T$. From central limit theorem considerations, the actual number of 1's is up to two leading orders $\widetilde{\mathbf{X}}_T$ is $p_X T + c_Z\sqrt{T}$ where c_Z will be a random normally distributed number, as long as T is large enough.

We create a graphical arrangement of *marks* using the two independent sequences on $[\![1, T]\!]^2$. A lattice square (i, j) is considered marked if and only if $X_i = Y_j = 1$. We also define a *lag* δ, $(0 \leq \delta \leq T)$; we are interested in the number of times the two processes obtained the value 1 in a time interval of size δ in either direction; in other words we want to know how many marks exist in a band of vertical height $2\delta+1$ around the main diagonal $D = \{(i, i) : 1 \leq i \leq T\}$.

In general, when there is availability of data, we can count the number of ones in the available time series (say n_X, n_Y) and then infer an approximation to the (generally unknown) parameters $p_X \approx n_X T^{-1}$ and $p_Y \approx n_Y T^{-1}$. Then one can run the theoretical model, as we describe it above, up to time horizon T and create two new independent time series, compare them with data and use them for predictions. This we use in Section 5.2.1, to derive an approximate central limit theorem (CLT) for the number of marks in a band, as $T \to \infty$. Part of the CLT is the expected value of marks of this completely independent model and approximate standard deviation. We use this result to construct a suitable Z-score for the number of marks in the band.

Consider a given lag δ and arrangements X, Y. We will find statistics for the number of marks in the band of distance δ around the main diagonal. The set of marks in the δ-band $D_{T,\delta} = \{(i,j) \in [\![1,T]\!]^2 : |i-j| \leq \delta\}$ is

$$S_{T,\delta} = \{(i,j) \in [\![1,T]\!]^2 : |i-j| \leq \delta, X_i = Y_j = 1\}. \tag{3.1}$$

We denote by $|S_{T,\delta}|$ the cardinality of the set $S_{T,\delta}$; this is precisely the number of marks in $D_{T,\delta}$. In the next two subsections we show the calculations for the expected value, limiting standard deviation and approximate normality for $|S_{T,\delta}|$.

For any statement A, we define the indicator function, $\mathbb{1}$, as:

$$\mathbb{1}\{A\} = \begin{cases} 1 & \text{if A is true} \\ 0 & \text{otherwise.} \end{cases}$$

3.2.2 Expected value of the number of points in a δ-band.

For the purposes of this section, fix parameters p_X and p_Y to denote the probability of success for two independent Bernoulli sequences $\{X_i\}_{i\geq 1}$ and $\{Y_j\}_{j\geq 1}$ so that each sequence is i.i.d. with marginal distributions

$$X_i \sim \text{Ber}(p_X) \quad \text{ and } Y_j \sim \text{Ber}(p_Y).$$

For any $T \in \mathbb{N}$, the random variable under consideration is $|S_{T,\delta}|$.

The computations rely on the decomposition

$$|S_{T,\delta}| = \sum_{(i,j):|i-j|\leq\delta} \mathbb{1}\{X_i = Y_j = 1\}.$$

With this we compute first the expected value of the number of marks in $D_{T,\delta}$.

$$\begin{aligned}
\mathbb{E}(|S_{T,\delta}|) &= \mathbb{E}\Big(\sum_{(i,j):|i-j|\le\delta} \mathbb{1}\{X_i = Y_j = 1\}\Big) = \sum_{(i,j):|i-j|\le\delta} \mathbb{E}\left(\mathbb{1}\{X_i = Y_j = 1\}\right) \\
&= \sum_{(i,j):|i-j|\le\delta} \mathbb{P}\{X_i = 1, Y_j = 1\} = \sum_{(i,j):|i-j|\le\delta} \mathbb{P}\{X_i = 1\}\mathbb{P}\{Y_j = 1\} \\
&= p_X p_Y \big(T(2\delta+1) - \delta(\delta+1)\big). \qquad (3.2)
\end{aligned}$$

At the last line of the computation above we used the number of terms in the sum. This can be calculated by adding the integer cells on the $2\delta + 1$ diagonals,

$$A_{T,\delta} = T + 2\sum_{k=1}^{\delta}(T-k) = T(2\delta+1) - \delta(\delta+1). \qquad (3.3)$$

Finally, observe that $A_{T,0} = T$, which is precisely the size of the main diagonal.

3.2.3 Central limit theorem for number of marks in δ-band as $T \to \infty$

We now present a different way to count the marks in the band, that will lend itself into the application of an ergodic CLT. For any $\delta + 1 \le i \le T - \delta - 1$, the total number of marks between $\delta + 1 \le i \le T - \delta - 1$ is given by

$$L_{T,\delta} = \sum_{i=\delta+1}^{T-\delta-1} X_i \sum_{j=i-\delta}^{i+\delta} Y_j. \qquad (3.4)$$

For each i, a direct calculation gives that

$$\mathbb{E}(X_i \sum_{j=i-\delta}^{i+\delta} Y_j) = \mathbb{E}(X_i)\mathbb{E}(\sum_{j=i-\delta}^{i+\delta} Y_j) = p_X p_Y(2\delta+1).$$

Define $W_i = X_i \sum_{j=i-\delta}^{i+\delta} Y_j$ for notational convenience.

Before stating the theorem, two observations follow. First we have the immediate inequality

$$L_{T,\delta} \le |S_{T,\delta}| \le L_{T,\delta} + (\delta+1)^2.$$

Thus, as $T \to \infty$, asymptotically $L_{T,\delta} \sim |S_{T,\delta}|$. The same inequality holds for expectations. Scaled by the same quantity, both variables satisfy the same limiting law as long as δ does not depend on T. If δ depends on T, one needs more refined arguments to show limiting results, and they will depend on this relation as well.

Second, notice that the random numbers $\{W_i\}_i$ have the same distribution for each fixed i and they are stationary and ergodic. Moreover, as long as $|i - k| > 2\delta + 1$, variables

$W_i = X_i \sum_{j=i-\delta}^{i+\delta} Y_j$ and $W_k = X_k \sum_{j=k-\delta}^{k+\delta} Y_j$ are completely independent. Therefore we have just defined a stationary ergodic sequence that is $2\delta+1$-dependent only. In particular this implies that the stationary sequence is *strongly mixing.*

We introduce some notation. First, let

$$Z_i = X_i \sum_{j=i-\delta}^{i+\delta} Y_j - p_X p_Y (2\delta+1) = W_i - \mathbb{E}(W_i).$$

Therefore $\mathbb{E}(Z_i) = 0$. Also note that $\mathbb{E}(Z_i^\ell) < \infty$ for any ℓ (high moment hypotheses are crucial for CLT for ergodic sequences). Then define (the value does not depend on the index ℓ below as long as $\ell \geq \delta + 1$)

$$\sigma_\delta^2 = \mathbb{E}(Z_\ell^2) + 2\sum_{k=1}^{2\delta+1} \mathbb{E}(Z_\ell Z_{\ell+k}) = \operatorname{Var}(W_\ell) + 2\sum_{k=1}^{2\delta+1} \operatorname{Cov}(W_\ell, W_{\ell+k}). \tag{3.5}$$

This variance is introduced because it is the one of $|S_{T,\delta}|$, see Theorem 3.2.2

Lemma 3.2.1. *The constant σ_δ^2 in* (3.5) *is given by*

$$\sigma_\delta^2 = (2\delta+1)p_X p_Y(1 - p_X p_Y) + 2\delta(2\delta+1)p_X p_Y(p_Y(1-p_X) + p_X(1-p_Y)). \tag{3.6}$$

Proof. We are going to compute each of the terms appearing in the last expression of (3.5). First,

$$\begin{aligned}
\operatorname{Var}(W_\ell) &= \operatorname{Cov}(W_\ell, W_\ell) = \operatorname{Cov}\Big(\sum_{j=-\delta}^{\delta} X_\ell Y_{\ell+j}, \sum_{k=-\delta}^{\delta} X_\ell Y_{\ell+k}\Big) \\
&= \sum_{j=-\delta}^{\delta}\sum_{k=-\delta}^{\delta} \operatorname{Cov}(X_\ell Y_{\ell+j}, X_\ell Y_{\ell+k}) = \sum_{j=-\delta}^{\delta}\sum_{k=-\delta}^{\delta} \big(\mathbb{E}(X_\ell^2 Y_{\ell+j} Y_{\ell+k}) - \mathbb{E}(X_\ell Y_{\ell+j})\mathbb{E}(X_\ell Y_{\ell+k})\big) \\
&= \sum_{i=-\delta}^{\delta} \big(\mathbb{E}(X_\ell^2 Y_{\ell+i}^2) - p_X^2 p_Y^2\big) + \sum_{j=-\delta}^{\delta}\sum_{k=-\delta, k\neq j}^{\delta} \big(\mathbb{E}(X_\ell^2 Y_{\ell+j} Y_{\ell+k}) - p_X^2 p_Y^2\big) \\
&= \sum_{i=-\delta}^{\delta} \big(p_X p_Y - p_X^2 p_Y^2\big) + \sum_{j=-\delta}^{\delta}\sum_{k=-\delta, k\neq j}^{\delta} \big(\mathbb{E}(X_\ell^2 Y_{\ell+j} Y_{\ell+k}) - p_X^2 p_Y^2\big) \\
&= \sum_{i=-\delta}^{\delta} \big(p_X p_Y - p_X^2 p_Y^2\big) + \sum_{j=-\delta}^{\delta}\sum_{k=-\delta, k\neq j}^{\delta} \big(p_X p_Y^2 - p_X^2 p_Y^2\big) \\
&= (2\delta+1)p_X p_Y(1 - p_X p_Y + 2\delta p_Y(1-p_X)).
\end{aligned}$$

The covariance for the other terms is computed in a similar way. First note that the variables common between W_ℓ and $W_{\ell+k}$ are the $Y_{\ell+j}$ for which $-\delta + k \leq j \leq \delta$ which precisely equal the values for $Y_{\ell+k+i}$ when $-\delta \le i \le \delta - k$. Then for any $k \in [\![1; 2\delta+1]\!]$

$$\begin{aligned}
\mathrm{Cov}(W_\ell, W_{\ell+k}) &= \mathrm{Cov}\Big(\sum_{j=-\delta}^{\delta} X_\ell Y_{\ell+j}, \sum_{i=-\delta}^{\delta} X_{\ell+k} Y_{\ell+k+i}\Big) \\
&= \sum_{j=-\delta}^{\delta}\sum_{i=-\delta}^{\delta} \mathbb{E}(X_\ell X_{\ell+k} Y_{\ell+j} Y_{\ell+k+i}) - \mathbb{E}(X_\ell Y_{\ell+j})\mathbb{E}(X_{\ell+k} Y_{\ell+k+i}) \\
&= p_X^2 \sum_{j=-\delta}^{\delta}\sum_{i=-\delta}^{\delta} \Big(\mathbb{E}(Y_{\ell+j} Y_{\ell+k+i}) - p_Y^2\Big) \\
&= p_X^2 \Bigg(\sum_{j=-\delta+k}^{\delta} \Big(\mathbb{E}(Y_{\ell+j} Y_{\ell+j}) - p_Y^2\Big) + \sum_{j=-\delta}^{\delta} \sum_{i=-\delta, i+k\neq j}^{\delta} \Big(\mathbb{E}(Y_{\ell+j} Y_{\ell+i+k}) - p_Y^2\Big)\Bigg) \\
&= (2\delta - k + 1) p_X^2 p_Y (1 - p_Y).
\end{aligned}$$

Then, overall

$$\begin{aligned}
\sigma_\delta^2 &= (2\delta+1) p_X p_Y (1 - p_X p_Y + 2\delta p_Y (1 - p_X)) + 2 p_X^2 p_Y (1 - p_Y) \sum_{k=1}^{2\delta+1} (2\delta - k + 1) \\
&= (2\delta+1) p_X p_Y (1 - p_X p_Y) + 2\delta(2\delta+1) p_X p_Y (p_Y(1-p_X) + p_X(1-p_Y)). \qquad \square
\end{aligned}$$

Then we can directly apply the central limit theorem for strongly mixing variables and obtain the following

THEOREM 3.2.2. *Let $S_{T,\delta}$ be the number of marks as defined by* (3.1). *Then, (with the notation introduced above), the following limit holds in distribution*

$$\lim_{T\to\infty} \frac{|S_{T,\delta}| - \mathbb{E}(|S_{T,\delta}|)}{\sqrt{T}} \overset{\mathcal{D}}{=} Z \sim \mathcal{N}(0, \sigma_\delta^2), \tag{3.7}$$

where σ_δ^2 is given by (3.6).

Proof. There exists a constant $C = C(\delta, p_X, p_Y) < \infty$ so that

$$\Big| \frac{|S_{T,\delta}| - \mathbb{E}(|S_{T,\delta}|)}{\sqrt{T}} - \frac{L_{T,\delta} - \mathbb{E}(L_{T,\delta})}{\sqrt{T}} \Big| \le \frac{C}{\sqrt{T}}.$$

This is a $\mathbb{P}$-a.s. statement, so as long as $T \to \infty$ the two fractions have the same distributional limit. Now focus on

$$\begin{aligned}
\frac{L_{T,\delta} - \mathbb{E}(L_{T,\delta})}{\sqrt{T}} &= \frac{\sum_{\delta+1}^{T-\delta-1} W_i - (T - 2\delta - 2)\mathbb{E}(W)}{\sqrt{T}} = \frac{\sum_{\delta+1}^{T-\delta-1} Z_i}{\sqrt{T}} \\
&= \frac{\sqrt{T - 2\delta - 1}}{\sqrt{T}} \cdot \frac{\sum_{\delta+1}^{T-\delta-1} Z_i}{\sqrt{T - 2\delta - 1}}.
\end{aligned}$$

As $T \to \infty$, the fraction multiplying the variable tends to 1 while the second fraction converges weakly to $\mathcal{N}(0, \sigma_\delta^2)$ by the CLT for strongly mixing ergodic sequences [14]. $\square$

3.3 Results

3.3.1 Agreement between analytical and empirical estimates

Given the strong dependence of the analytical expressions on the product of the rates ($p_X p_Y$), we examined the agreement between analytical and empirical estimates for i.i.d. uniform point processes along two dimensions – event density and rate homogeneity – considering both average and limit cases. Concretely, we considered 4 pairs of point processes emitting events in $[\![1;T]\!]$, T=1,000, with rates (0.05, 0.05), (0.05, 0.75), (0.75, 0.75) and (0.25, 0.50), respectively. Note that $p_X = p_Y = 0.75$ was chosen because it is the limit (for large lags) of the rate at which variance is maximised in the homogeneous case. As shown by Figure 3.1, agreement between analytical and empirical estimates is excellent across all scenarios, even when the expected number of events is extremely low. It should be noted that whereas the expected value of the statistic is insensitive to rate inhomogeneity, i.e., the expected value for one pair of point processes emitting (1,100) events and that of a pair emitting (10,10) events will be identical, the variance is not. For a given product of rates, the more heterogeneous the rates, the higher the standard deviation.

Next, we investigated the extent to which agreement between analytical and empirical estimates holds when considering considering scenarios that depart from the assumptions underlying our analytical derivation. First, we considered the case of sampled (or binned) point processes, as is frequently the case in many real-world scenarios, e.g., in physiology. Next, we introduced auto-correlations in the point processes as this has been shown to induce biases when measuring statistical dependence between point processes, e.g., when considering Poisson processes [120].

Impact of sampling frequency

We investigated the impact of sub-sampling both in independent Bernoulli sequences and in homogeneous Poisson processes. First we considered the case of independent Bernoulli sequences. Starting from reference signals of length T (i.e., T time bins of length 1), we constructed sub-sampled sequences of length $\lceil \frac{T}{L} \rceil$ where each time bin contained a 1 if there was one or more event in the corresponding L time bins in the original sequences, 0 otherwise. Note that where L was not a divisor of T, the last time bin was smaller than L. The effect of this was considered negligible. Sub-sampling has the effect of increasing the effective rate of each sequence. Since each time bin remains independent, the new rate can

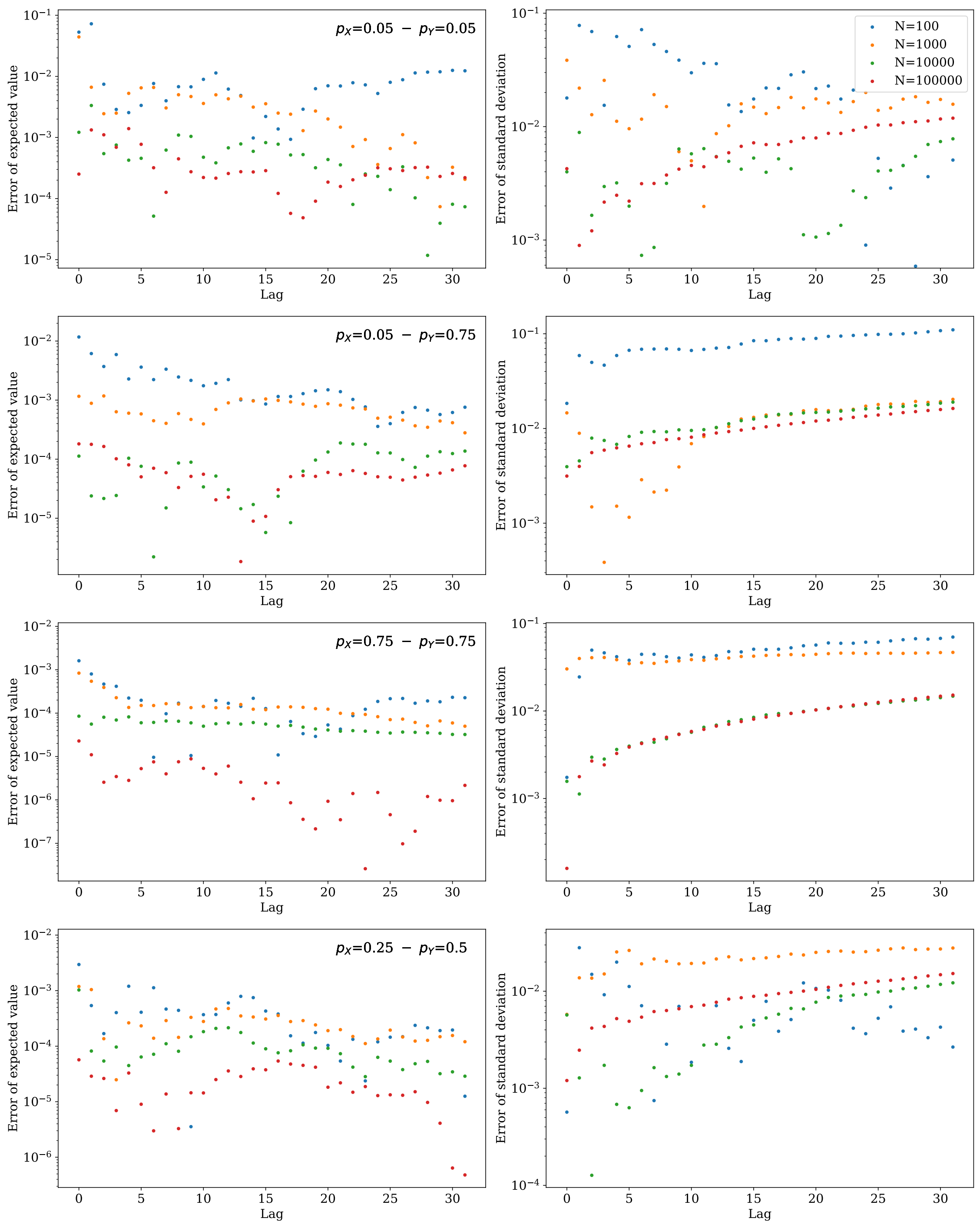


Figure 3.1: Normalised difference between the analytical and empirical expected value (left panels) and standard deviation (right panels) as a function of the lag for different values of p_X and p_Y. The normalised difference between analytical variable (expected value or standard deviation) V_a and empirical variable V_e is calculated as $\frac{|V_a - V_e|}{V_a}$. The empirical expected value and standard deviation are estimated using N=10^2, 10^3, 10^4 and 10^5 samples. Results show that with N=100,000, both analytical expected values and analytical standard deviations are always less than 1% away from their empirical counterpart.

be calculated as $1-(1-p)^L$ and therefore the analytical formulas we derived for expected value and standard deviation should hold with those rates.

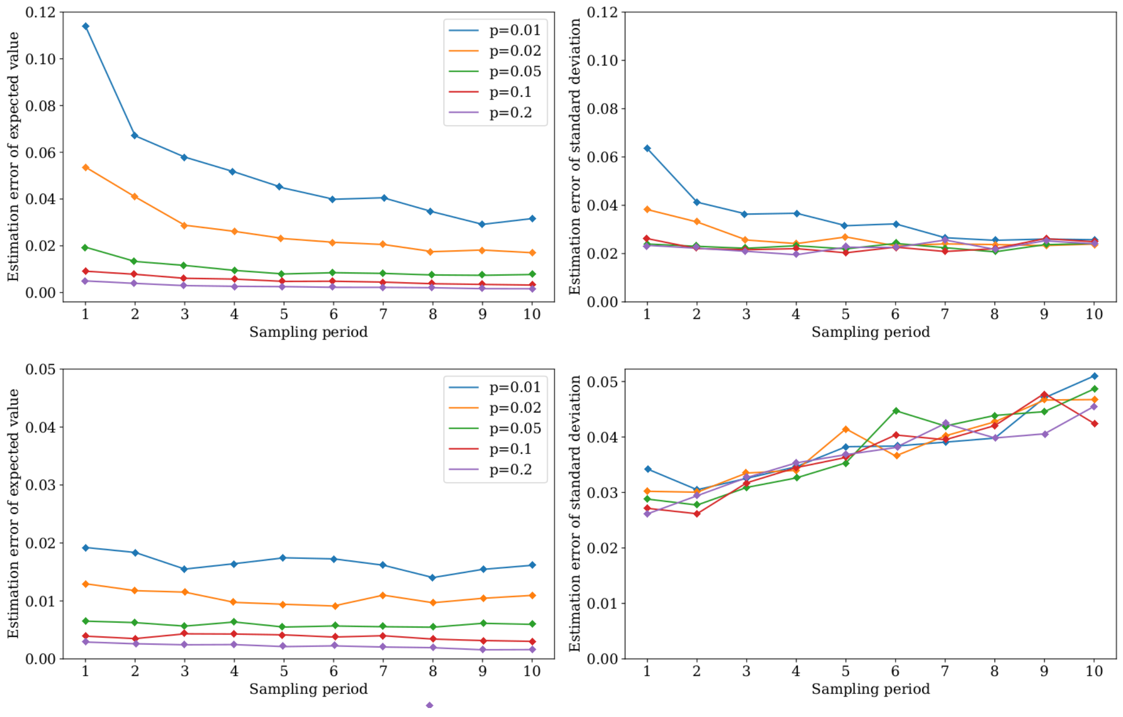


Figure 3.2: Normalised root-mean-square error as a function of the sampling period L for both expected value (left) and standard deviation (right) at lag 0 (top row) and lag $\sqrt{\frac{T}{L}}$ for Bernoulli sequences with rates provided in the legend.One hundred sets of 1,000 pairs of sequences were used. That is to say that we estimated hundred times both the expected value and the standard deviation using 1,000 pairs of point processes and that we computed the normalised root-mean-square error using those hundred estimates. For analytical variable (expected value or standard deviation) V_a and empirical variable V_e, it is equal to $\frac{1}{V_a}\sqrt{\frac{1}{100}\sum_{i=1}^{100}(V_a-V_e)^2}$. The non-sampled sequences had length $T=1,000$, that is to say that before sub-sampling the sequences are made of T time stamps and that after sub-sampling, with sampling period L, they are made of $\lceil\frac{T}{L}\rceil$ bins.

This is confirmed by Figure 3.2 which shows small normalised root-mean-square errors (NRMSE) for both quantities, irrespective of the choice of lag and sampling period. For both small (top row) and large (bottom row) lags, estimation of the expected value is shown to be sensitive to the rate p, improving with higher rate. This is expected because given the finite (fairly small, in this case) length of the sequence, the smaller the rate, the less robust the estimation is. For the same reason, it is observed that at low lag (here, 0, i.e., instantaneous cross-correlation; top row, left panel), the error decreases with increasing sampling period, with this improvement increasingly pronounced for decreasing rates. This is because with larger bin sizes, the resulting rate $1-(1-p)^L$ increases, leading to more robust estimations. For sufficiently high rates and sampling period (e.g., $p = 0.2$, sampling period of 3; same panel), the error becomes insensitive to the choice of sampling period. Qualitatively similar observations can be made for the estimation of the standard deviation (top row, right panel). For large lags (bottom row), the estimation error of the expected value is insensitive to the choice of sampling period as expected from the fact that at large lags, the number of events allows for a more robust estimate. Further, it is significantly reduced compared to that at low lag (e.g., error for the smallest rate below 0.02 at all sampling periods, compared to up to 0.12 at small lag; bottom row, left panel). The estimation error on the standard deviation is insensitive to the value of the rate but marginally increases with the sampling period (bottom row, right panel). This is expected from the fact that the rates on the basis of which the standard deviation is calculated are an increasingly poorer approximation of the actual rates following sampling.

Next, we considered the case of homogeneous Poisson processes in lieu of Bernoulli sequences. For a given time horizon and rates, homogeneous Poisson processes will emit events that satisfy our assumptions provided the sampling frequency is large enough and therefore we should expect excellent agreement between analytical and empirical estimates. However, if the sampling frequency is reduced (or put differently, the rates increase in relation to the sampling frequency), then the likelihood that two events fall within the same time bin increases. For a Poisson process of rate λ the probability that there are n events within any bin of duration b is:

$$P(N(t+b)-N(t)=n)=\frac{(\lambda b)^n}{n!}e^{-\lambda b} \tag{3.8}$$

And therefore, the expected number of events 'lost' to binning over the given time horizon T is:

$$\mathbb{E}(N_L)=\frac{T}{b}\sum_{n=2}^{\infty}(n-1)\frac{(\lambda b)^n}{n!}e^{-\lambda b}=\lambda T-\frac{T}{b}(1-e^{-\lambda b}) \tag{3.9}$$

This shows that for a given time horizon T and known expected number of events, our statistic should be more accurate if the sampling frequency is high (i.e., the bin size b is small). To briefly illustrate this, we investigated agreement when the sampling period was kept constant (unit time bins) but the rates were increased such that the fraction of events lost to binning varied between 0.5% and 20%. For the sake of brevity, we once again only considered processes with homogeneous rates and quantified agreement using NRMSE at lags 0 and $\sqrt{T}$.

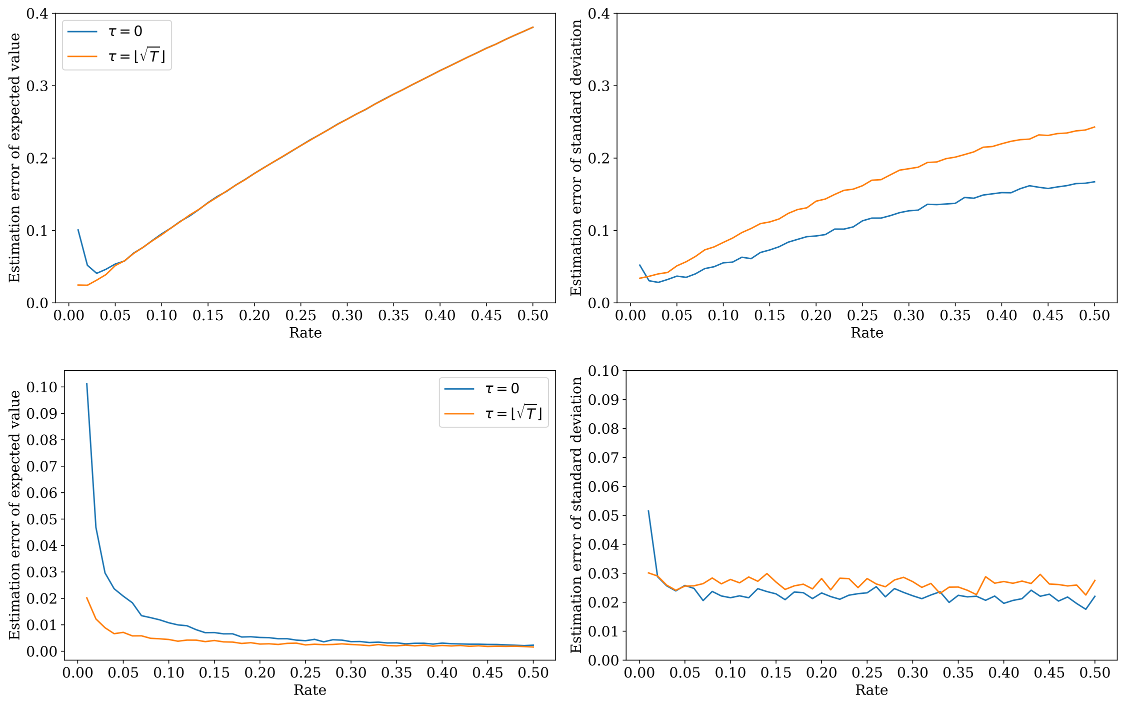


Figure 3.3: RMSE for both expected value (left) and standard deviation (right) at lags 0 and $\sqrt{T}$ for homogeneous Poisson processes with varying rates. Top: Error when the analytical value is calculated assuming no events were lost through binning. Bottom: Error when the rate was estimated based on the observed number of events. One hundred sets of 1,000 pairs of sequences were used.

As shown by Figure 3.3(top), when not correcting the rates for the loss of events through binning, both estimation errors increase with larger rates, i.e., as the proportion of events lost to binning increases; and this degradation is rapid, with a 20% loss of events resulting in a RMSE of almost 40% in the expected value. This error is insensitive to the lag considered, except at very low rates when a low number of events (at lag 0) leads to an unreliable estimation process as explained earlier. When correcting the rates for the loss of events, however, Figure 3.3(bottom) shows that except for very low rates, there is very good agreement across all rates, as demonstrated in appendix 3.5.1.

Auto-correlation

We used geometric AR(1) processes [101] to generate auto-correlated waiting times for the events of the point processes. The geometric AR(1) process is defined as:

$$X_t = \alpha * X_{t-1} + B_t G_t \tag{3.10}$$

where X is a discrete random variable Geometric with parameter $0 < \theta < 1$, $\{B_t\}$ and $\{G_t\}$ are independent sequences of iid random variables such that the random variable B_t is binary with parameter $0 \leq \alpha \leq 1$ and the random variable G_t is Geometric with parameter θ, and the $*$ operator is defined as:

$$\alpha * X_t = \sum_{i=0}^{X_t} Y_i \tag{3.11}$$

where $\{Y_i\}$ is a sequence of iid Binary random variables with parameter α. Put simply, this model states that the elements at time t, X_t, are the sum of (1) the survivors at time $t-1$, X_{t-1}, each with probability of survival α and of (2) elements from the innovation process B_tG_t. For any non negative integer k, the auto-correlation at lag k is given by $\rho(k) = \alpha^k$. In particular, α is the auto-correlation at lag 1.

To quantify the effect of auto-correlations on the agreement between analytical and empirical estimates, we generated point processes systematically varying α for processes with rate $p \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$ over $[\![1; T]\!]$, T=1,000. Agreement was once again quantified in terms of the normalised root-mean-square error (see legend of 3.2 for details). As shown by Figure 3.4(top row), the estimation error in both expected value and standard deviation increases with the degree of auto-correlation. This is expected because introduction of auto-correlation leads to a departing of the structure of the inter-events intervals from the one underlying the derivation of our statistic (Bernoulli processes). Strikingly, we observe that for all lags, the estimation errors increase as the rates decrease. This can be explained as follows. For a fixed horizon T, the larger the rate, the less variability there is in the structure of the inter-events intervals. Hence, introduction of regularity within the distribution inter-events intervals via introduction of auto-correlation, has a greater impact for lower rates. For low levels of correlation (< 0.2) and sufficiently high rates (> 0.05), there is good agreement between analytical and empirical estimates of the expected value. In fact, as shown by the bottom row of Figure 3.4, for $\alpha = 0.1$, there is very good agreement for both expected value and standard deviation over all lags even in the worst case scenario of $p = 0.01$ (where the expected number of events is only 10).

3.3.2 Normal convergence of the empirical estimates

In Section 5.2.1, we have shown that in the limit of large horizon times T, the distribution of our statistic converges to a normal distribution with parameters the analytical

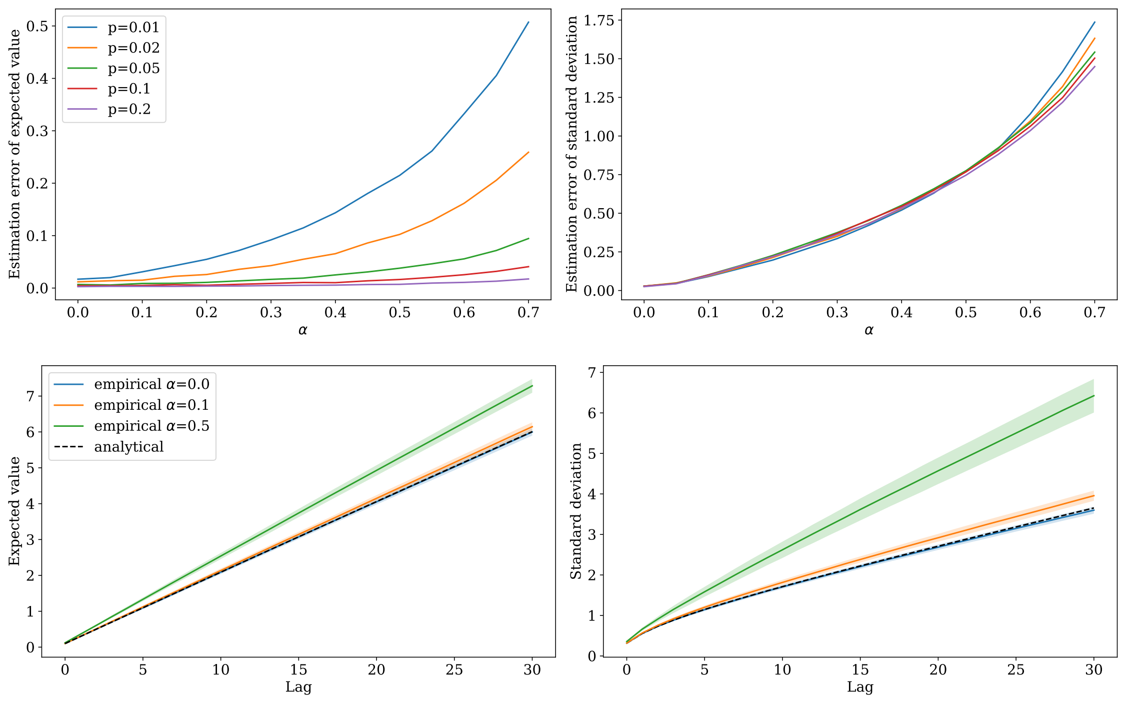


Figure 3.4: Top row: Normalised root-mean-square estimation error of the expected value (left panel) and standard deviation (right panel) as a function of α (the auto-correlation at lag 1) at lag $\sqrt{T}$ for the 5 rates shown in the legend. Those errors were calculated over over 100,000 estimates. Bottom row: Empirical expected value (left panel) and standard deviation (right panel) as a function of the lag for $p = 0.01$ (the worst case scenario) and three values of auto-correlation $\alpha \in \{0, 0.1, 0.5\}$. The analytical curves (black dotted line) are provided for reference. Shaded areas denote the standard deviation of the empirical estimates and were computed over one hundred sets of 1,000 estimates.

results provided in (3.2) for the expected value and (3.6) for the variance. By way of empirical validation, we systematically varied rates and lags and showed that there was always a data duration after which the distribution of the empirical estimates could be said to be normally distributed based on the evidence of a KS test [2] with $p > .05$ (with 5000 estimates). Figure 3.5 shows those times for the rates and lags considered. Since the expected value is only dependent on the product of the rates, we used homogeneous rates, $\lambda_x = \lambda_y = \frac{n}{T} \in \{0.01, 0.02, ..., 0.5\}$. Focusing on homogeneous rates is justified by the fact that convergence is only marginally affected by heterogeneity, as demonstrated by panel D. The gradual convergence of the empirical distributions to a normal distribution is illustrated in panels B and C when the lag and recording duration increase, all other

[2]Normality was also assessed using both Shapiro Wilk and Anderson Darling tests and results were consistent across all three tests (data not shown).

factors being kept constant.

Empirical convergence to a normal distribution is significantly impacted by the fact that our statistic takes integer values (it is a count of co-occurrent events). First, normal approximation can only be achieved if there is a wide range of values the statistic can take. This is clearly evidenced by panels B and C whereby the p-value increases (the distribution becomes more normal) as recording duration (respectively lag) increases leading to a wider range of values. Using kernel density estimation (yellow curve) provides a smoother estimation of the density which matches closely the analytical one (red), see right-hand side panels for example, however, the empirical distribution is step-like and therefore not assessed as being normal. Second, if the statistic at a lag is close to 0 or close to its maximum value ($n_X n_Y$), the integer-valued nature of the statistic will once again prevent the empirical distribution to be approximated as normal. For example, the distribution of the statistic in the first plot in panel B is not symmetric because of its lower bound. This was previously noted by [116]. More generally, this issue is illustrated by the top row of Figure 3.5 whereby estimates for low rate processes require significantly longer durations for the empirical distributions to become approximately normal.

The asymptotic convergence of the statistic around its expected value justifies the construction and use of a Z-score to measure deviation from the assumption of independence. For two time series X and Y emitting n_X and n_Y events within $[\![1;T]\!]$, it is defined as:

$$Z_{X,Y}(\tau) = \frac{|S_{T,\tau}| - \mathbb{E}(|S_{T,\tau}|)}{\sigma_\delta \sqrt{T}} \tag{3.12}$$

where $|S_{T,\tau}|$ is the statistic at lag τ, $\mathbb{E}(|S_{T,\tau}|)$ is the analytical expected value (3.2) and σ_δ the analytical standard deviation (3.5) where we are estimating parameters p_X, p_Y with their empirical estimators $\frac{n_X}{T}$ and $\frac{n_Y}{T}$ respectively.

3.3.3 Application to a delayed version of the Common Shock Model

Description of model

The bivariate Poisson Common Shock Model provides a flexible platform to validate the effectiveness of our statistic in quantifying pairwise coupling between two point processes. Indeed it is often used as a benchmark for bivariate counts [54]. In its classic form, and following [29], it takes two Poisson point processes $X_1 \sim \mathcal{P}(\lambda_{X_1})$ and $X_2 \sim \mathcal{P}(\lambda_{X_2})$ and assumes that each of them can be written as the sum of two independent Poisson

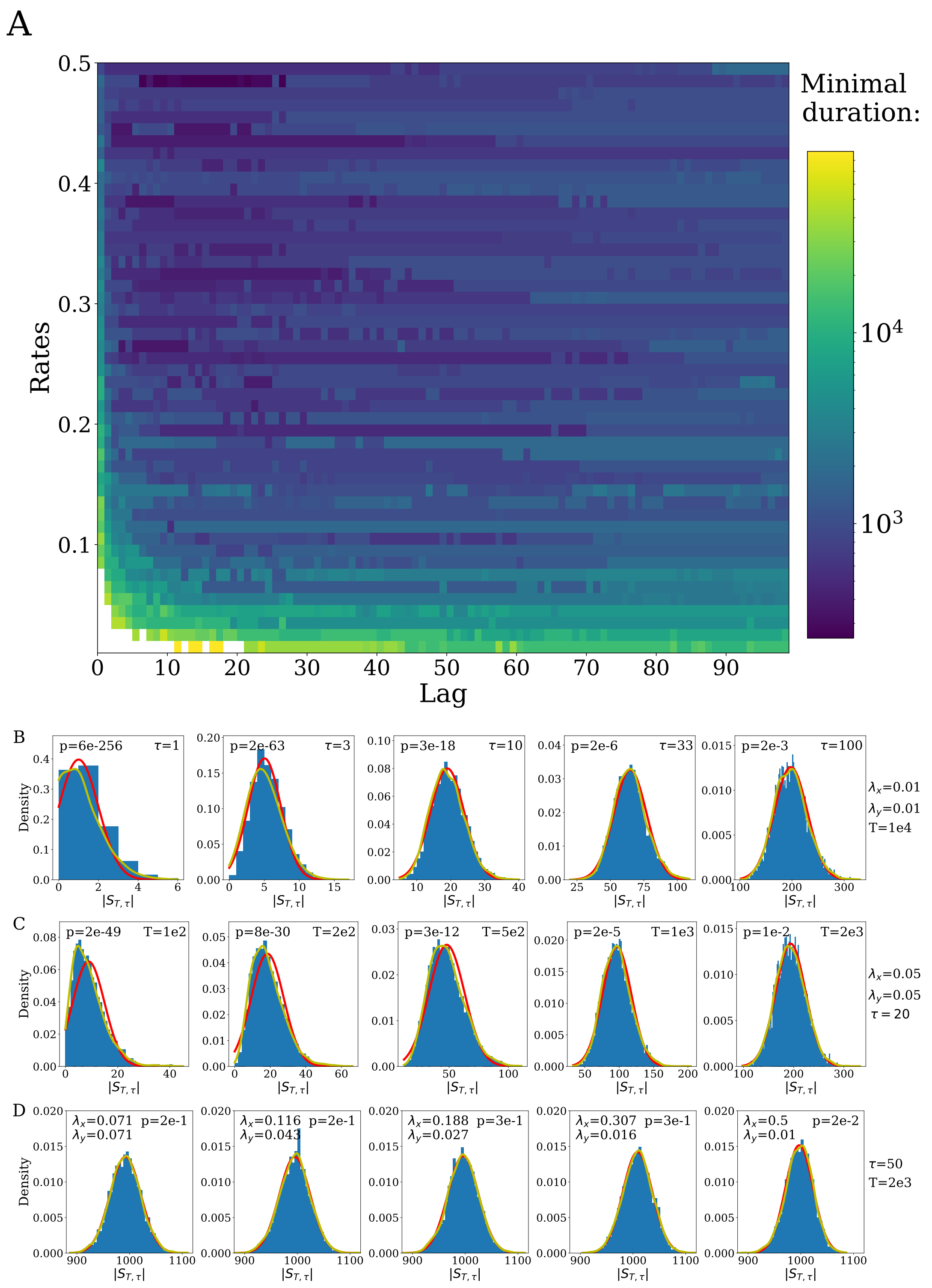


Figure 3.5: **A:** Minimal recording duration after which the empirical distribution is approximately normal as a function of rates and lags. **B:** Empirical distributions as a function of the lag for given total duration and rates. **C:** Empirical distributions as a function of the total duration for given lags and rates. **D:** Empirical distributions as a function of rate heterogeneity for given total duration, lags and rate products. All panels: (red) the theoretical distribution for the scenario; (yellow) the Gaussian kernel density estimation of the empirical distribution .

processes respectively Z and Y_1, and Z and Y_2, where $Z \sim \mathcal{P}(\lambda_Z)$, $Y_1 \sim \mathcal{P}(\lambda_{X_1} - \lambda_Z)$ and $Y_2 \sim \mathcal{P}(\lambda_{X_2} - \lambda_Z)$. Whilst the choice of rates λ_Z, λ_{Y_1} and λ_{Y_2} enables to adjust the strength of the interaction, in this form, the model only allows for instantaneous interaction. Here, we use the delayed version [36] such that, instead of Z, we consider two processes Z_1 and Z_2 that are delayed versions of Z, with delays taken from some probability distribution function $F_i(.), i \in \{1, 2\}$ (see Figure 3.6 for a graphical representation). This process has a number of possible interpretations but a very prominent one in the neurophysiology literature is the notion of *common drive* [136]. Since Z is not actually observed and we are not interested in recovering it, without loss of generality, it is possible to consider that only one of the process, e.g. Z_2, is a delayed version of Z, the other, Z_1 being equal to Z. We thus denote $Z^\star$ the delayed version of Z. To model the delay we used a normal distribution of parameters μ_δ and σ_δ, allowing us to control both average lag between events from Z and $Z^\star$ and lag jitter.

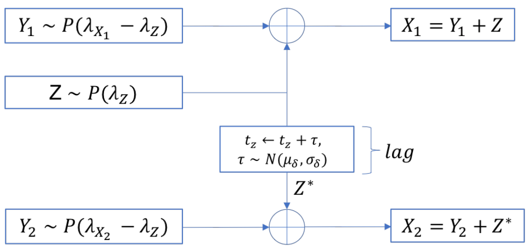

$Y_1 \sim P(\lambda_{X_1} - \lambda_Z)$
$X_1 = Y_1 + Z$
$Z \sim P(\lambda_Z)$
$t_z \leftarrow t_z + \tau,$
$\tau \sim N(\mu_\delta, \sigma_\delta)$
lag
Z^*
$Y_2 \sim P(\lambda_{X_2} - \lambda_Z)$
$X_2 = Y_2 + Z^*$

Figure 3.6: The Delayed Common Shock Model

Brief validation of the statistic

Since previous results from Section 3.3.1 showed excellent agreement between empirical and analytical estimates of the expected value and standard deviation of the statistic for independent Poisson processes, application of the Z-score to the delayed common shock model in the no-interaction case (i.e., $\lambda_z = 0$) should lead to a Z-score taking an expected value of 0 and a standard deviation of 1. This behaviour is confirmed by the left panel of Figure 3.7 with consistent empirical estimates (expected value and standard deviation) across all lag values.

As representative example of the delayed interaction case, we considered the case where all 3 processes had the same rate $\lambda = 0.01$, an average lag $\mu_\delta = 50$ and jitter $\sigma_\delta = 10$ (chosen to be significantly smaller than the average lag, as expected in most real-world scenarios). The behaviour of the Z-score is shown in right panel of Figure 3.7. At first sight, the interpretation of this behaviour is not straightforward. First, it is observed that whilst the Z-score for value of lags $\tau \ll \mu_\delta$ is close to zero, it is not zero. This is a direct implication of the delay introduced by the model. The analytical expected value of the statistic provides the expected number of co-incident events at each lag conditional to the processes emitting a given number of events over the time horizon T. With delayed interaction, and depending on the relationship between the delay and the rate of the processes, the actual number of co-incident events observed for lags less than the delay will diverge from that expected at random. In this case, with each process only emitting (on average) 100 events (i.e., 1 every 100 time steps), a delay of 50 is likely to reduce the likelihood of co-occurrence at very small lags. The Z-score then shows a steady increase peaking at around $\mu_\delta + 1.5\sigma_\delta$ before steadily decreasing. The fact that the Z-score does not peak at the average lag but later is a natural result of the cumulated nature of our statistic. To illustrate this, we calculated the derivative of the Z-score (orange line) as a proxy for the cross-correlation at each lag and superimposed the distribution of delays used in the model (green line). As the lag approaches the mean lag, the number of co-incident events begins to exceed that expected at random and therefore, the Z-score increases. The rate of increase is maximum where the distribution of delays peaks. As the lag increases from the mean delay, the ratio between the number of observed co-incident events to the number of co-incident events expected at random reduces and with it the Z-score. Once again, because the Z-score is a cumulative statistic (measuring the number of co-incident events up to the lag considered), the statistic will not return to 0. In other words, the Z-score at high lags takes into account the fact there has been substantial interaction at smaller lags. There are two implications to this. First, the Z-score will not show great sensitivity to interactions that have large (in relation to the duration of the record) average lags. Second, a positive value of the Z-score at a given lag does not indicate that significant interaction is actually taking place at that lag. Our results are based on calculating the statistic at each of all lags considered. Note that the alternative approach of extracting the peak value of the statistic over those lags for each realisation would not permit comparison to a $\mathcal{N}(0,1)$ distribution as noted in [89] for cross-correlation analysis. For such comparison to be made, one needs to look at each lag. However, given the cumulated nature of the Z-score, any substantial interaction will lead to significant Z-score values over a wide range of lags (greater than the mean delay) such that any prior knowledge and/or hypothesis as to the nature of the delay (e.g., conduction velocity in neurophysiology) could be usefully exploited.

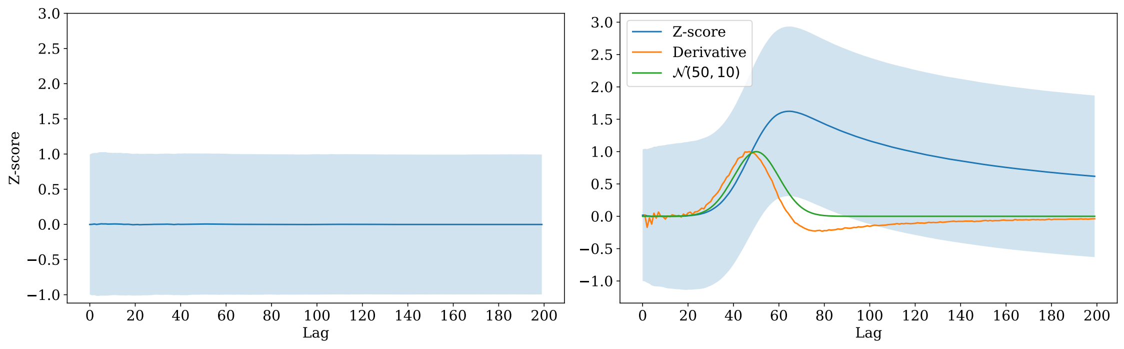


Figure 3.7: Estimated expected value of the Z-score over 10,000 trials as a function of the lag in absence (left) and presence (right) of delayed interaction. We used T=10,000, $\lambda_{y1} = \lambda_{y2} = 0.01$ and $\lambda_z = 0$ for the left panel; T=1,000, $\lambda_{y1} = \lambda_{y2} = \lambda_z = 0.01$, $\mu_\delta = 50$ and $\sigma_\delta = 10$ for the right panel.

3.4 Discussion

In this chapter, we have proposed a statistic that enables the robust assessment of the presence of (instantaneous or delayed) statistical dependence between two point processes. This statistic extends rigorous work on Pearson's correlation [81] by providing rigorous results when considering interaction over multiple lags. The identification of statistical dependence is based on an exact derivation of the number (and standard deviation) of co-incident events expected to be observed over a time period given two discrete independent point processes. This statistic depends only on parameters that can be readily estimated by counting the number of events observed over a given period of time and is valid for every value of those parameters, even extreme cases (e.g., only one event over the entire record), unlike its frequency-domain counterpart, coherence. A central limit theorem for this number was derived that demonstrates convergence of the distribution of the statistic to a normal distribution of known parameters and permits the construction of a Z-score quantifying the likelihood of a given number of co-incident events in a time period being observed if the processes were independent.

Agreement between analytical and empirical values was verified and the statistic was shown to be well behaved even when departing from the assumptions of the model. For example, it was shown to be robust to sampling (in most reasonable cases) as well as when auto-correlated point processes were considered (within limit). Furthermore, we showed that this Z-score is more adequate to identify statistical dependencies than the classically used coherence when recording are short and information sparse.

The statistic has two main limitations. First, from an interpretation viewpoint, it is somewhat less intuitive than other methods due to its cumulative nature. A significant Z-score at a particular lag does not imply significant underlying dependence at that lag, merely, that there was dependence at one or many shorter lags. This means some care must be taken when selecting the lag(s) over which to make an inference. Nevertheless, we have shown that in some cases, it will be possible to make inference about the nature of the interaction through differencing of the method. Second, in its basic form, and like most such techniques, the statistic assumes stationarity of the signals. Adaptations of the statistic to handle non-stationarity (e.g., via optimal filtering or windowing) is left for future work.

3.5 Appendix

3.5.1 The case of Bins for Poisson processes

Suppose that N_X, N_Y are independent Poisson processes with rates λ_X, and λ_Y respectively. We will discretise these processes using bins of size $b > 0$, with a time horizon T.

Partition the time interval $[0, T]$ into disjoint intervals of size b. In other words, interval k (denoted by I_k) is the interval $[(k-1)b, kb)$, for $1 \leq k \leq [Tb^{-1}]$. The Poisson processes are independent in each interval, so we define a sequence of independent Bernoulli variables

$$X_i = \mathbb{1}\{N_X(ib) - N_X((i-1)b) \geq 1\}, \quad \text{and} \quad Y_j = \mathbb{1}\{N_Y(ib) - N_Y((i-1)b) \geq 1\} \tag{3.13}$$

Their respective probabilities of success are

$$p_X = 1 - e^{-\lambda_X b} \quad \text{and} \quad p_Y = 1 - e^{-\lambda_Y b}. \tag{3.14}$$

Therefore, using the bin size b, we reduce this to our discrete model, with discrete time horizon $T_d(b) = [Tb^{-1}]$ and success probabilities for the Bernoulli variables. The lag value δ from the continuous model is also scaled by the bin size; since it needs to be an integer, by convention we take it to be $\delta_d(b) = [\delta b^{-1}]$.

Using this information, the expected number of marks will be $\mathbb{E}(|\widetilde{S}_{T_d(b),\delta_d(b)}|)$ and the variance needed for the CLT will be of leading order $\sigma^2_{\delta_d(b)} T_d(b)$. Using these theoretical results, one can quantify the estimation error induced by binning according to b, since we will be ignoring all events but 1 that happen in the same sub-interval I_k.

Suppose for convenience that $\lambda_X \geq \lambda_Y$. The probability $p_{\lambda_X,b}$ of seeing 2 or more Poisson points for N_X in I_k is

$$p_{\lambda_X,b} = 1 - e^{-\lambda_X b}(1 + \lambda_X b) \leq 1 - (1 - \lambda_X b)(1 + \lambda b) = (\lambda_X b)^2.$$

Therefore, if $b \ll \lambda_X^{-1}$ then this probability will be small.

Furthermore, if $\lambda_X b \leq CT^{-\alpha}$ for some $\alpha > 1/2$, then a Taylor expansion of the expected value of points lost because of binning shows that $\mathbb{E}(N_L) \ll \sqrt{T}$ which is not enough to alter the leading order of $\sigma_\delta^2\sqrt{T}$ and $\mathbb{E}(|\widetilde{S}_{T,\delta}|)$. In this case the Poisson model and the discrete model obtained by decreasing bin sizes will coincide.

3.6 Annex

NB: This section does not belong to the paper submitted to PRE and provides a deeper analysis of the performance of the Z-score using the delayed common shock and as well as a comparison to two standard coupling measures.

3.6.1 Sensitivity analysis

The delayed common shock model enables us to model a pairwise coupling and control several of its characteristics: (1) by modifying μ_δ, we can change the average coupling lag, (2) by modifying σ_δ, we can change the lag jitter, (3) by multiplying λ_z, λ_{y1} and λ_{y2} by a unique scalar, we can modify the event density, (4) by modifying the relationship between λ_z and $(\lambda_{y1}, \lambda_{y2})$, we can change the coupling intensity and (5) by modifying T, we can change the number of observable data points, i.e. the number of observable periods. The purpose of this section is to provide the reader with an understanding of the behaviour of the Z-score to different kind of couplings. To analyse the impact of each of the coupling characteristic, we consider a reference setting and we modify one or more of the parameters while keeping the others constant. For reasons explained throughout this analysis, the main one being that changing the parameters do not actually modify the general shape of the Z-score curve, we select as reference setting: $T = 10,000$, $\lambda_{y1} = \lambda_{y2} = \lambda_z = 0.01$, $\mu_\delta = 30$ and $\sigma_\delta = 5$. We plot the results on Figure 3.8.

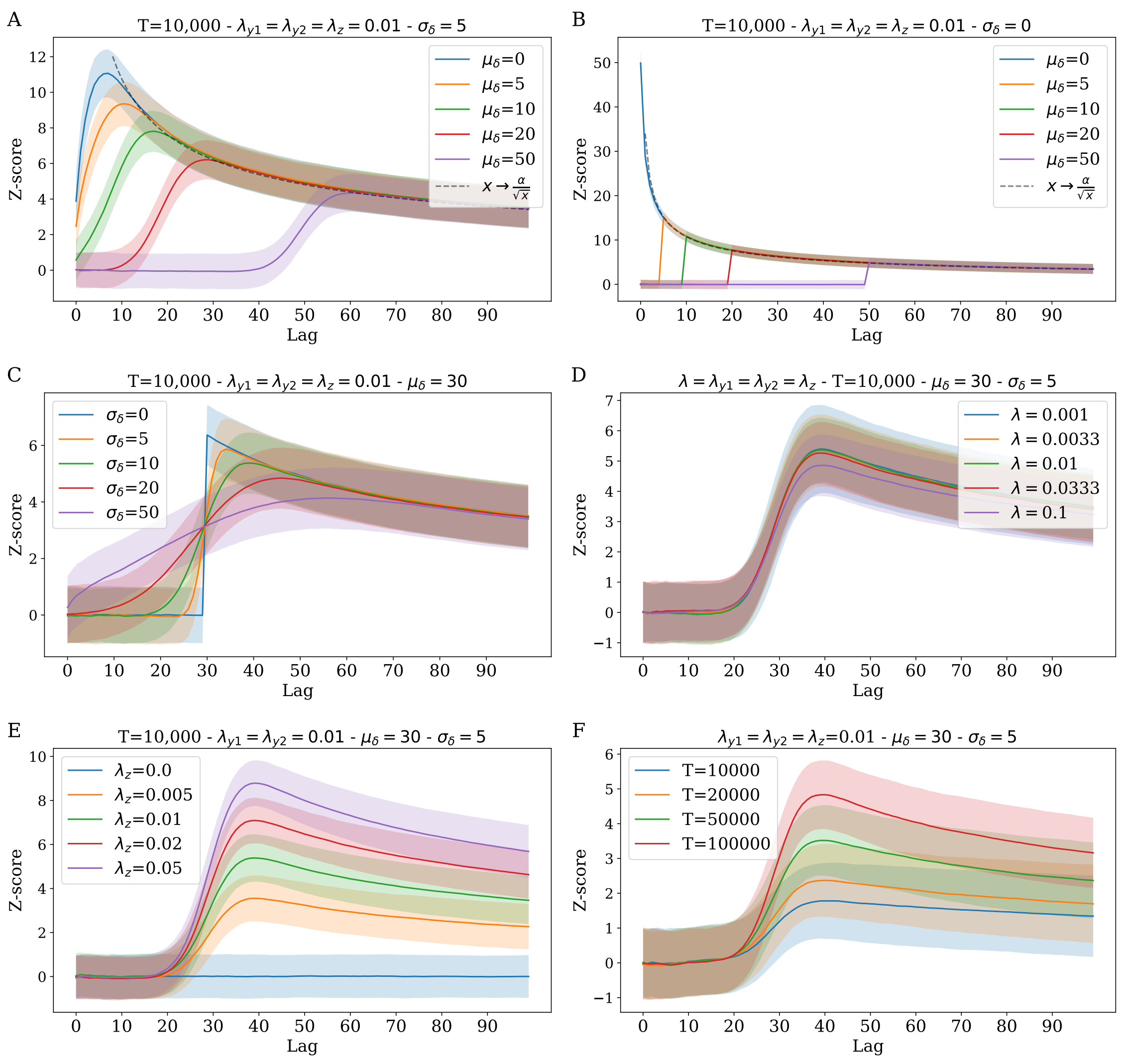


Figure 3.8: Evolution of the average over 1,000 trials of the Z-score as a function of the lag for different kind of coupling. The non changing parameters are written on the top of Figures while the changing one are written in the legend box. **A-B:** Impact of the mean average lag with and without jitter. A curve in $x \rightarrow \frac{\alpha}{\sqrt{x}}$, $\alpha \in \mathbb{R}$ is fitted. **C:** Impact of the jitter. **D:** Impact of the density. **E:** Impact of the intensity. **F:** Impact of the number of data points observable assuming constant rates.

Average lag: As expected and revealed by the top two panels of Figure 3.8, the larger the average interaction lag is, the smaller the maximal value of the Z-score is, and hence the less likely an interaction is to be detected with the Z-score. Besides, we observe that the different Z-scores merge a few lags after the average coupling lag. This is neither a surprise. Indeed, these scenarios differ only by their average coupling lag. Hence the dynamics should be the same long before and long after the average coupling lag. And this is what we observe: before the Z-score starts to increase and approximately $1.5\sigma_\delta$ time stamps after the average interaction lag, all the curves are merged. Finally, we remark that the Z-score decreases with the inverse square root of the lag as shown by the fitted dotted curve. The fit is not perfect but helps quantifying the decrease of the Z-score after the average coupling lag.

Lag jitter: Introducing jitter within the interaction smoothes the shape of the Z-score and decreases its maximal value. This is to be expected for two reasons. First, when jitter is introduced, the coupling stops to be restricted to a unique lag but spreads around it. Second, all things being equal, x "surprising" pair of events are less surprising if observed up to lag l_1 than up to lag $l_2 < l_1$. Next, as in the previous two panels, we remark that the curves merge for large delays. This happens because couplings differ only by their lag jitter. Finally, we remark that the Z-score peaks approximately around $\mu_\delta + 1.5\sigma_\mu$, c.f. section 3.3.3 for a qualitative explanation. This implied that the stronger the jitter is, the later the Z-score peaks. Since the Z-score decreases with the square root of the lag, we understand why the Z-score peak value decreases with the jitter.

Interaction density: The D panel of Figure 3.8 suggests that as long as the ratios $\frac{\lambda_{y1}}{\lambda_z}$ and $\frac{\lambda_{y2}}{\lambda_z}$ remain constant and equal to each other, then the Z-score is only weakly dependent of the numerators and denominators. Although across the studied scenarios the coupling intensity may vary of two orders of magnitude, the Z-score varies only by a few percent. This suggests that, unlike methods that require a lot of data points, the Z-score is well suited to detect presence of coupling from sparse time-series of events.

Interaction intensity and Number of observable points: Unsurprisingly, the more coupled pair of events there are, the easier it is to detect the presence of interaction, see panel E. This is easy to understand, the more coupled pairs, the easier it is to observe a pattern and hence the more confident one becomes about the presence of a statistical dependency. Note that when $\lambda_z = 0$ the Z-score is on average equal to 0 for all lags and that otherwise all Z-scores have the same shape. Likewise, it should not be surprising that the longer the observation, the easier it is to detect the presence of a coupling, see panel F. This because as the recording duration increases, the more confident it is also possible to become about the presence of a pattern. Note also that all the Z-scores peak at approximately the same lag on panel E and F. This is interesting because it suggests that this peak does only depend of the the jitter and that we can expect to find it approximately at lag $\mu_\delta + 1.5\sigma_\mu$.

3.6.2 Performance comparison

In this section, we compare the performance of the Z-score against the one of two standard coupling measures: Pearson's cross-correlation and coherence [69]. From the understanding gained from previous analysis of those two standard measures [23], we aim at showing that the Z-score is filling up a gap, that there are some conditions for which the Z-score is more efficient at detecting the presence of a delayed interactions than both other standard coupling measures. We start by explaining how we use the two standard methods. The interpretation of coherence is straightforward. It has well known analytical confidence intervals. The coherence $C_{X,Y}(\lambda)$ of two independent processes X and Y for any frequency λ has a probability equal to $\alpha \in [0;1]$ to be larger than $1 - (1-\alpha)^{\frac{1}{L-1}}$, where L is the number of non overlapping windows used to estimate the coherence [68]. In the following we use *scipy* to estimate the coherence using non-overlapping windows made of $2^8 = 256$ time steps. This choice is motivated by the fact that it is the smallest power of 2 that is at least twice as large as the maximal interaction lag studied. To interpret the output of the cross-correlation a reference is needed and surrogate data are hence often used [141, 180]. Recent works developed surrogate data that preserve the inter-event intervals and the non-stationarity of the data [97]. However we know that the uncoupled sequences generated by the delayed version of the common shock model are stationary and emit uniformly distributed events. As for surrogate data, we will thus generate sequences using the delayed version of the common shock model with $\lambda_z = 0$.

In order to show that there exist some conditions for which the Z-score is best able to detect the presence of existing interaction, we start by building expectations based on the knowledge we gained from the previously performed sensitivity analysis and from previous analysis of both coherence and Person's cross-correlation [23, 68]. First, note that the analytical expression of the confidence interval of the coherence reveals that coherence is very dependent on the number of observable windows. For instance, in the worst case scenario, i.e. when only one window is used, the coherence is constantly equal to 1 and no measure can thus be made. We hence expect the coherence to perform poorly if little data are available. Second, the cross-correlation measures statistical dependency at every lag, hence if this dependency is spread across many lags, the cross-correlation is likely to perform worse. However the cross-correlation is independent of the average coupling lag, hence we can expect it to be better at detecting coupling at large lags. Third, since (1) the Z-score decreases with inverse of the square root of the lag after the detection of an interaction and (2) it peaks around $\mu_\delta + 1.5\sigma_\mu$, it should perform poorly if the average interaction lag or the jitter are very large. Based on those analysis, it is possible to build expectations:

- First, the Z-score should be better, than Pearson's cross-correlation, at detecting the presence of coupling if the average coupling lag is small and the jitter is large. The reciprocal proposition is also expected.
- Second, we expect the Z-score to be better than the coherence at detecting the presence of coupling if the recording duration is small.

In the following we test those predictions.

Comparison to Pearson's cross-correlation: Since at lag 0, the Z-score is measuring exactly the same thing as Pearson's cross-correlation, they should be equally able to detect the presence of an interaction for such a lag. This hence suggests that if we want to measure the effect of the average lag and jitter, the choice of the recording duration, i.e. T, and of the different rates, i.e. λ_{y1}, λ_{y2}, and λ_z should not be decisive. Thus we assign to those parameters the values chosen in section 3.6.1. The average lag and jitter remained to be chosen. We consider two scenarios. Within the first one, the average lag is small and jitter is large. Within, the second one, the average lag is large and the jitter is small. Many possible values of the average lag and jitter were possible and we show representative results on Figure 3.9.

When the coupling jitter is large and average coupling lag is small as on the top panel of Figure 3.9, Pearson's cross-correlation, compared to the Z-score, struggles to detect the presence of coupling. The distribution of the cross-correlation and of the Z-score suggest

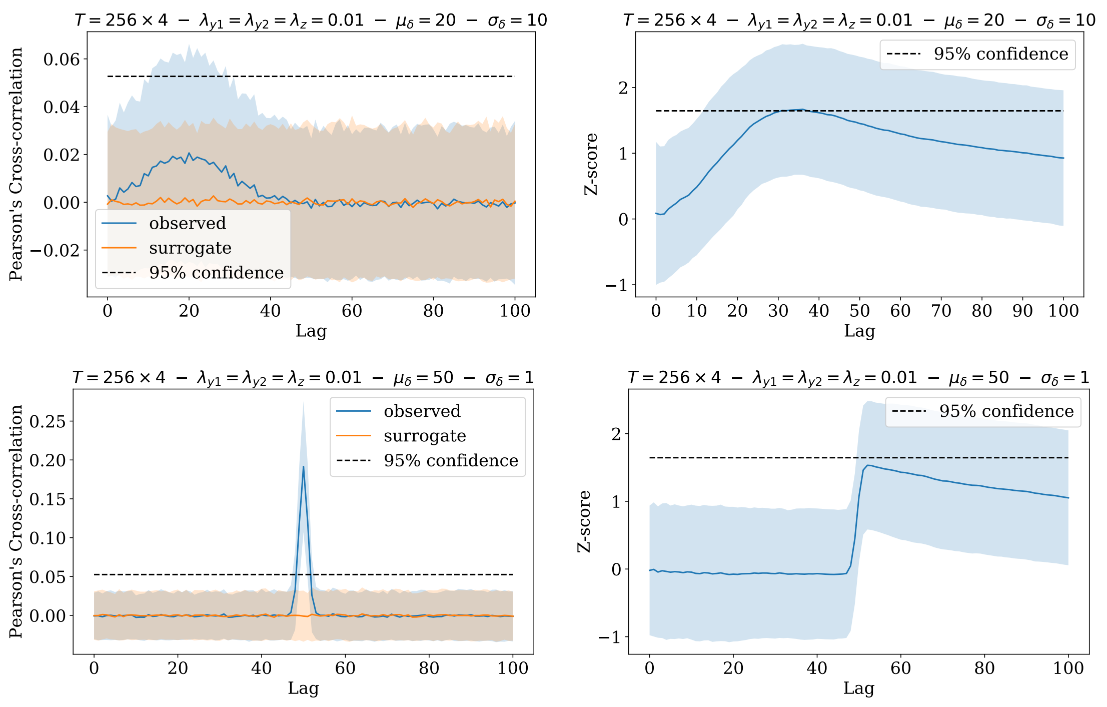


Figure 3.9: Comparison between Pearson's cross-correlation and the Z-score: the Z-score is better suited to detect jittered coupling at short lag while Pearson's cross-correlation is better suited to detect the presence of delayed and little jittered coupling. Evolution of Pearson's cross-correlation (left) and of Z-score (right) as a function of the lag. The parametrisation are described on the top of each panel. The standard deviations are computed using 1,000 samples. The Z-score 95% confidence line is obtained based on the assumption of a $\mathcal{N}(0, 1)$ distribution. The cross-correlation 95% confidence line is obtained by assuming that the cross-correlation of the surrogate data is normally distributed.

indeed that the use of the cross-correlation leads less often to the detection of an existing coupling. This is because the cross-correlation measures the coupling at each lag and is not able to combine them unlike the Z-score which is a cumulative function. On the contrary, as revealed by the lower panel of Figure 3.9, Pearson's cross-correlation is more efficient than the Z-score at detecting the presence of a coupling if the jitter is small and the average lag is large.Together those findings provide confirmative evidences of our expectations and suggest that there exist indeed settings in which it would be more appropriate to use the Z-score than Pearson's cross-correlation.

Comparison to coherence: In order to maintain some consistency with the previous analysis, we use the same reference scenario as in Figure 3.9 and consider the ability of both the Z-score and the coherence to detect the presence of coupling for two different recording durations. We plot the results on Figure 3.10.

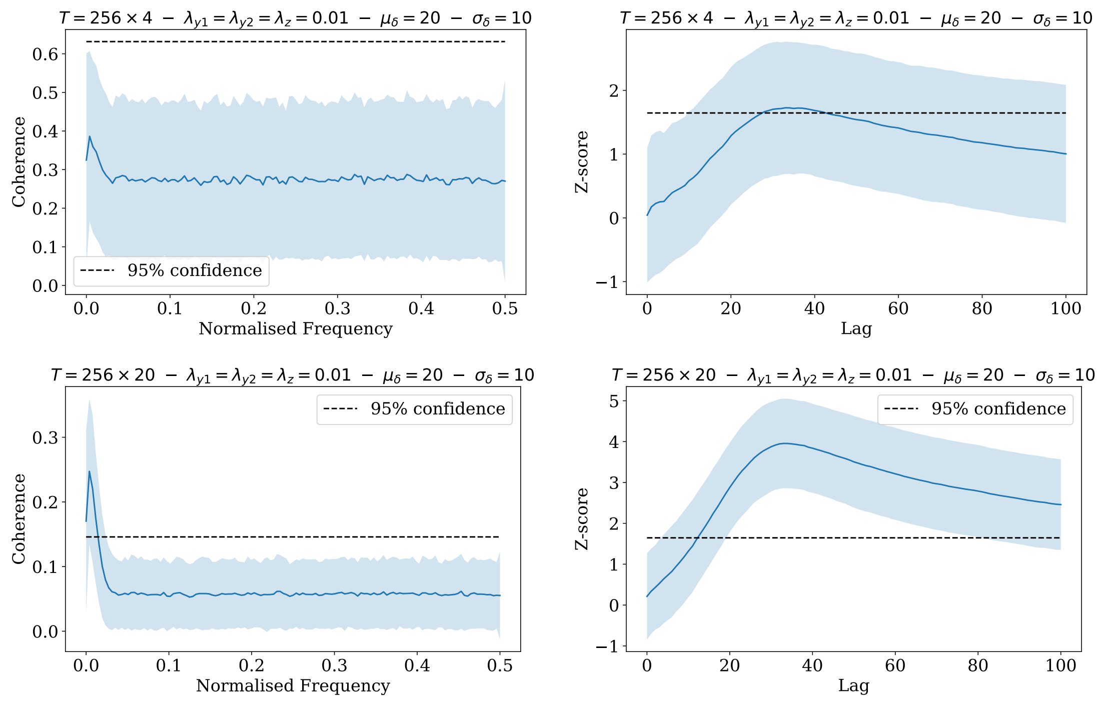


Figure 3.10: Comparison of the coherence and the Z-score: the Z-score is less dependent than the coherence on the recording duration. Evolution of the coherence as a function of the normalised frequency (left) Evolution of the Z-score as a function of the lag (right). Settings are described at the top of each picture. The standard deviations are computed using 1,000 samples. The Z-score 95% confidence line is obtained based on the assumption of a $\mathcal{N}(0, 1)$ distribution.

First note that when the recording duration is small, the coherence is almost not able to detect the presence of coupling. It is indeed only when the recording duration encompass many windows, e.g. 20, that there exists at least one frequency for which we can detect the presence of an interaction with more than 95% confidence. This suggests that the Z-score is much less dependent on the recording duration than the coherence and that it is better suited to detect the presence of an interaction for small recordings. Finally note that the scenario examined on the top of Figure 3.10 is the same as the one of the top of Figure 3.9. This strongly suggests that there exist at least one type of interaction for which the Z-score is more suited to detect than both coherence and Pearson's cross-correlation.

Chapter 4

A novel way to estimate edge probability from the distribution of network coupling statistics

NB: This chapter has not been submitted or pre-submitted anywhere.

Abstract: *Networks are a powerful tool to understand the dynamics of complex systems. Their connectivity is however often lacking and needs hence to be inferred from the activity of each of their node. Because of their simplicity and computational efficiency, pairwise statistics are still commonly used to perform such inference. The connectivity is then often extracted from the thresholding of such statistics. Statistically most principled approaches choose a threshold that limits the false positive rate via false discovery rate procedure. We argue that network inference thresholding approaches would greatly benefit from procedures that also take false negative rates into account. In specific contexts, the distribution of pairwise statistics calculated from the activity of independent pair of nodes, are approximately known. In this chapter, we assume that the statistics of independent pair of nodes are normally distributed (although this could be generalised to other distribution), and develop a method that aims at reconstructing the distribution of independent pair of nodes' statistics. It outputs the size, the parameter of this distribution and the function that associates to each pair of nodes' statistics the probability that this pair of nodes is actually connected. We then experimentally demonstrate the performance of the method using a mixture of normal and/or approximately normal distributions that account for dependent and independent pair of nodes statistics. Further, using the same mixtures, we show our method is almost always able to recover the connectivity with a higher F1 score than approaches based* *false discovery rate procedure. Finally we highlight the limitations of the proposed method, propose ways to make it more resilient and conclude by stressing the fact that there is an unused and exploitable informational value in the non-edge distribution that would make analysis based on inferred network more robust.*

4.1 Introduction

Networks are becoming a very common tool to understand complex interactions that many studied systems exhibit. Through a binary conceptualisation of the reality, they are able to model individuals and their relationships via sets of nodes and edges. For instance, in neuroscience nodes and edges may account for neurones and axons [145]. The simplicity and generality of such representation, combined with an increasing number of theoretical tools, provided many important scientific advances in many different fields [110]. However since interactions between individuals may not be directly observable, most of the network representations are not readily available. Connectivity between nodes must hence often be inferred from the activity of each node. Depending on the nature of the activity (e.g. continuous or discrete), on the desired nature of the retrieved interaction (e.g. functional or causal), on the nature of the system studied (e.g. neural spikes, computer events, gene expression level, tweets) many different tools have been developed and their efficiency depends on the context [23, 25].

Although interactions often encompass simultaneously more than a pair of nodes, many of the approaches used today to infer the network connectivity still rely, because of their simplicity and efficiency, on pairwise coupling measures [141, 122]. Those tools output a weighted matrices where each weight $w_{i,j}$ is associated with a quantity related to the probability and/or the strength of the interaction between nodes i and j. However, in case of independence the weights output by most coupling measures are not strictly equal to 0 but rather tend to be taken from a non null distribution. For instance after a Fisher transform, the Pearson's cross-correlation of two independent normal variables approximately follows a normal distribution centred on 0 and of variance $\frac{1}{T-3}$, where T is the recording duration [47]. Hence, many weights associated to independent nodes may be large by chance and inferring the network connectivity from the weighted matrix is not straightforward.

The simplest and most common solution to overcome the aforementioned issues consists in thresholding the weighted matrix [140]. This involves binarising the weighted matrix by linking nodes i and j if and only if their weight, $w_{i,j}$, is larger than a given threshold. However the choice of such threshold is difficult and no consensus has been reached. Thresholding approaches can be assigned into two categories: fixed threshold and fixed network feature threshold [140]. The aim of all approaches is to maintain consistency accross trials and individuals. Methods belonging to this first category usually either choose to limit the false positive rate via a false discovery rate procedures [11, 89] or employ a predefined unique threshold across all trials and individuals [104]. Both approaches are questionable. On the one hand, choosing a priori a false positive rate is problematic because it prevents any control on the false negative rate. On the other hand, choosing a priori a threshold is even more arbitrary since the binarised network comes with no confidence level. Both approaches have hence often led to an incorrect description of studied systems [52]. The second category aims at preserving a feature of the structure of the binary inferred networks across all trials and individuals. Many features have been proposed: the number of separated components [42], the overall density of edges [52, 159], the average degree [10] or some other complicated features [73]. This list is not exhaustive but its length highlights the lack of consensus. All of these approaches have different advantages but are all based on strong assumptions about the structure of the networks that may not always be justified and may thus lead as well to inconsistent results [52]. Hence analysis avoiding thresholding have gained traction. It usually consists in performing the analysis on the whole weighted matrix instead of a sparse binary one [86, 56]. However such solutions do not overcome the challenge posed by the potential existence of spurious edges. While they effectively give a lower weight to edges that are less likely to exist, they do not discard them.

In this chapter, we propose to a way to overcome the thresholding issue by estimating the probability that each weight is actually associated to an interacting pair of nodes. This procedure assumes that weights associated to independent pair of nodes are distributed according a normal distribution of unknown parameters (although other distributions could easily be selected) and aims at recovering this distribution from the mixture of weights associated to dependent and independent pair of nodes. The method outputs a function that associates to each weight the probability that it is associated to an independent pair of nodes. This function can then easily be used to recover the binary network with the best F1 score or the weighted network where most unlikely edges are discarded. In section 4.2, we explain how the method works. Then in section 4.3.1, using two normal distributions of known parameters, accounting for independent and dependent pairs of nodes, we describe in depth how the proposed procedure works and provide evidence for its correct behaviour. Then, in section 4.3.2, we consider different normal and approximately normal distributions and perform a sensitivity analysis. Next in section 4.3.3, we use use the same dataset and compare the F1 score provided by our procedure with the one provided by false discovery rate approaches. Then in section 4.3.4, we apply our method to the dataset described in chapter 2. Finally in section 4.4, we analyse the potential impact of this work, highlight its limitation and provide future directions.

4.2 Method

Although this approach could be generalised to different distributions, since under specific conditions, the coherence and the cross-correlation of two independent processes tend to be normally distributed [47, 69], the proposed method focuses on normal distributions. The weights of the weight matrices W output by the chosen method (e.g. coherence, correlation) can theoretically be distributed among two categories: (1) the weights, w^e, associated to actually existing edges (simply called edges weights in the following) and (2) the weights, w^n, associated to non existing edges (later called non-edge weights). Two points are important to note. First, edge weights follow an unknown distribution while non-edge weights are assumed to follow a normal distribution of a priori unknown parameters μ_n and σ_n. Second, by definition of the coupling measures, non-edge weights are expected to be smaller than edges weights. The proposed method can be divided into two parts. First it infers the number of edges and the parameters μ_n and σ_n and second it estimates the function that assigns edge probabilities to weight values.

Estimation of the parameters: The aim of this first part is to infer from the weight samples $W = \{w_i, i \in [\![1; N]\!]\}$ the number of non-edges, N_n and the two parameters μ_n and σ_n. From the two remarks made earlier, we expect the proportion of edge weights among weights smaller than a threshold to decrease as the value of the threshold decreases. However since we cannot make any assumption about the shape and the size of the edge weights sample, there is a heuristic trade-off to make. It can be summarised as follows. On the one hand as the threshold decreases, the distribution modeling weights smaller than this threshold tends to be closer to a truncated normal distribution (the proportion of weights associated to non edges increases), but on the other hand less observations are available and inferences become harder to make. We thus expect that there is a threshold that best enables to estimate the distribution of the non-edges weights. To find this best threshold, we proceed in a grid like manner. First we select an arbitrary number of possible thresholds covering the whole range of weights. Next, for each threshold, using a maximal likelihood approach, we fit a truncated normal distribution, of estimated mean and variance $\hat{\mu}_n$ and $\hat{\sigma}_n^2$, to the thresholded data (see Annex). We then measure the goodness of fit using the Kolmogorov-Smirnov (KS) distance between the thresholded data and the fitted truncated normal distribution[1]. And we finally select the threshold t (and the associated estimated parameters $\hat{\mu}_n$ and $\hat{\sigma}_n$) that minimise the KS distance. Hence, assuming that the proportion of edge weight is negligible within the thresholded data (compared to the proportion of non-edge weight), we propose to estimate the number of non-edge N_n as:

$$\hat{N_n} = \frac{\sum_{i=1}^{N} \mathbb{1}(w_i \leqslant t)}{F_{\hat{\mu}_n,\hat{\sigma}_n}(t)}, \tag{4.1}$$

where $N = N_n + N_e$ is the total number of weights, $w_i,\ i \in \{1,..,N\}$ are the observed weights, $F_{\hat{\mu}_n,\hat{\sigma}_n}(t)$ is the cumulative distribution function of a $\mathcal{N}(\hat{\mu}_n, \hat{\sigma}_n)$ random variable and $\mathbb{1}(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$. Note that this threshold is not the one that can be used to discriminate between edges and non-edges and to binarise the network, see Figure 4.1.

Estimation of the probabilities: Given the estimated number of non-edges ($\hat{N_n}$) and the parameters of the normal distribution (μ_n and σ_n), the aim of this second part is to transcribe the distribution to an edge probability distribution. Given a weight interval centred in w and of length dw, $[w - \frac{dw}{2}; w + \frac{dw}{2}]$, it is possible to estimate the expected number of non-edge weights within this interval:

$$n_{exp}(w, dw) = N_n \bigg(F_{\hat{\mu}_n,\hat{\sigma}_n}(w + \frac{dw}{2}) - F_{\mu_n,\sigma_n}(w - \frac{dw}{2}) \bigg), \tag{4.2}$$

where $F_{\hat{\mu}_n,\hat{\sigma}_n}(t)$ is the cumulative distribution function of a $\mathcal{N}(\hat{\mu}_n, \hat{\sigma}_n)$ random variable. However the number of observed weights $n_{obs}(w, dw)$ within the interval $[w - \frac{dw}{2}; w + \frac{dw}{2}]$ is often going to be different from $n_{exp}(w, dw)$. Indeed, as illustrated on Figure 4.1 for weights w larger than the threshold t, there is a non negligible proportion of edge weight.

We assume that the difference, i.e. $n_{obs}(w, dw) - n_{exp}(w, dw)$, is equal to the number of edge weights. Within $[w - \frac{dw}{2}; w + \frac{dw}{2}]$, we hence estimate the proportion of edge weights as:

[1] We use the KS distance because when used to measure the distance between an analytical probability distribution function, e.g. truncated normal one, and an empirical distribution, the distance increases as the size of empirical distribution decreases. When used as an optimisation metric, it hence favors larger observed distributions.

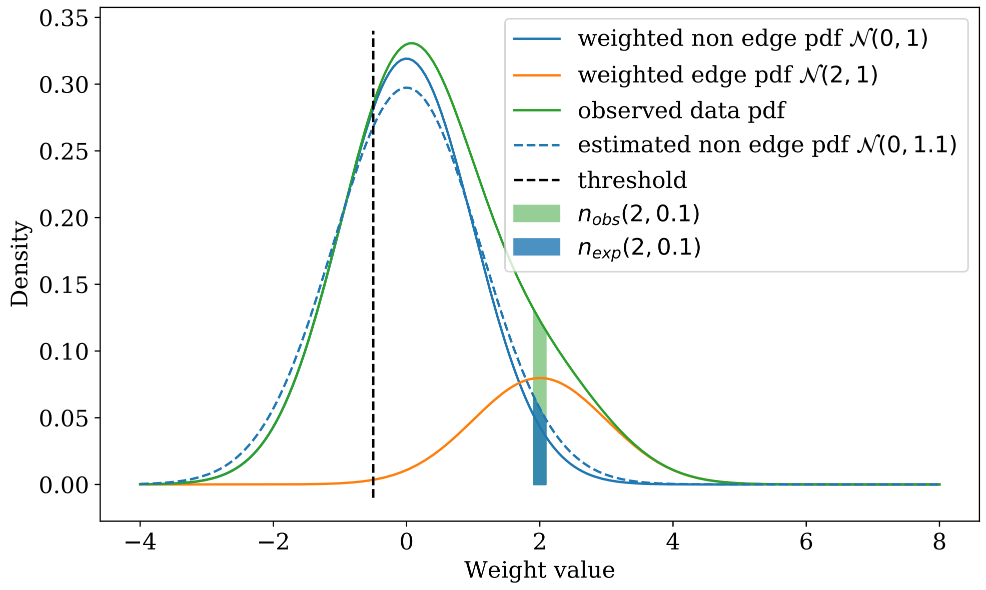


Figure 4.1: Schematic representation of the method. The scenario represented here considers a scenario in which 80% of the weights are non-edges (blue line) and follow a $\mathcal{N}(0,1)$ distribution and 80% of the weights are edges (orange line) and follow a $\mathcal{N}(2,1)$ distribution. Note that what is originally observed is the aggregation of these two distributions (green line). For illustration purpose, we assume that the first part of the method led to the selection of a threshold $t = -0.5$ (black dashed line) and found that the non edge weights follow a $\mathcal{N}(0,1.1)$ distribution (dashed blue line). The hence expected number of non edge weights within $[1.9; 2.1]$ is proportional to the light blue area, while the number of observed ones is proportional to the light green area.

$$\hat{p}(w,dw) = \frac{n_{obs}(w,dw) - n_{exp}(w,dw)}{n_{obs}(w,dw)}. \tag{4.3}$$

However, because of ineluctable fluctuations, this proportion may be negative. Hence to transform this proportion into a probability we propose to fit a sigmoid function to it. The sigmoid is defined as:

$$f_{\alpha,\beta}(w) = \frac{1}{1+e^{\alpha+\beta w}}, \tag{4.4}$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are parameters to be found using for instance mean square error estimate. We chose such a function because we observed that it models well the cumulative distribution function of many distributions. Once α and β are found, the function $f_{\alpha,\beta}$ associates to each weight w, our estimation of the probability that the weight w is associated to an edge.

4.3 Results

4.3.1 First validation and in-depth illustration of the method:

In order to show that the proposed method is effectively able to estimate edge probabilities from the weight samples, we first analyse its performance in a scenario for which ground truth is known. Hence, we consider two weight distributions associated with edges and non-edges. The non-edge one, is normal by assumption. To simplify we also make the edge distribution normal. Divergences from normality are studied in the sensitivity analysis section 4.3.2. Each distribution is described by 3 parameters: their expected value, $\mu_{e/n}$, their standard deviation, $\sigma_{e/n}$ and their size, $N_{e/n}$. Without loss of generality we can fix the non-edge distribution parameters to $\mu_n = 0$ and $\sigma_n = 1$. As can be expected and as shown in section 4.3.2, the more the distributions overlap the harder it is to recover each distribution. For illustration purpose, we consider distributions that sufficiently overlap so that they cannot be readily distinguished from one another and that are sufficiently far away so that inference is possible. We hence propose to consider the set of weights that would result from the observation of a network of 100 nodes with 10% edge density and where edge weights follow a normal distribution of parameters μ_e and σ_e equal to 3 and 1, see Figures 4.2 and 4.3.

Figure 4.2 illustrates the first part of the inference process. For different threshold values, using the maximum likelihood approach we infer the parameters ($\hat{\mu}_n$ and $\hat{\sigma}_n$) that best explain the thresholded, i.e. truncated, data and plot them in green (expected value) and red (standard deviation). We observe that as the threshold decreases, their values decrease as well until they stabilise (and later fluctuate) around their theoretical values as represented by the black dotted lines. This decrease is to be expected. Indeed for large thresholds, the truncated distribution is made of both edge and non edge weights. Since edge weights tend by nature to be larger, their incorporation necessarily increases the average weight value. This explains why the inferred parameters ($\hat{\mu}_n$ and $\hat{\sigma}_n$) are larger than expected for larger thresholds.

The proposed method uses the Kolmogorov-Smirnov (KS) distance (in blue) to measure the goodness of fit between the observed truncated distribution and the theoretical truncated normal distribution of parameters the one predicted by the maximum likelihood approach. At first, i.e. for large threshold, the KS distance decreases as the threshold decreases. This suggests a bad fit for large thresholds. This is to be expected. For large

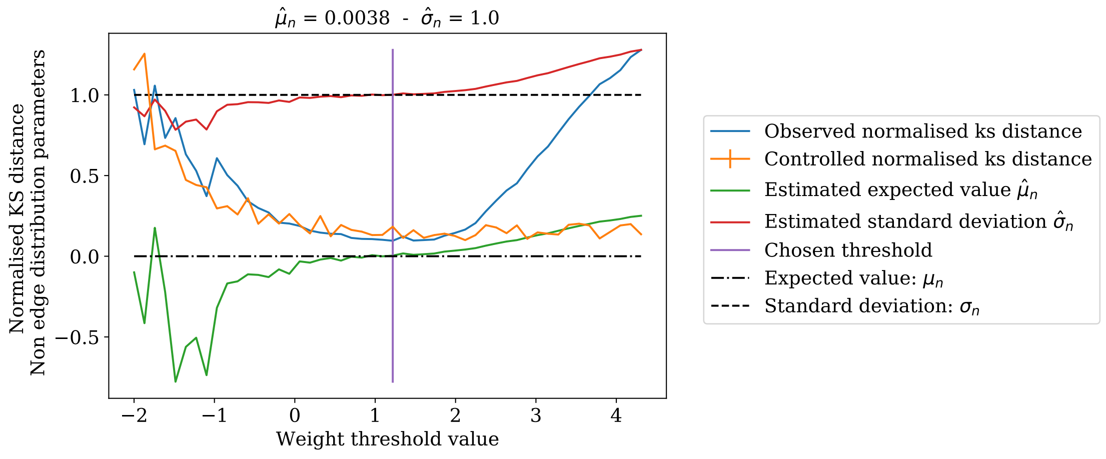


Figure 4.2: Illustration of the first part of the inferring process. Different variables are plotted as the threshold, used to estimate the truncated normal distribution, changes. The inferred parameters of the truncated distributions are in red and green, their true value are shown by black dotted lines. In blue, we plot the normalised KS distance between: (1) the observed truncated data and (2) a theoretical normal truncated distribution of parameters the estimated ones (in green and red). The minimal KS distance value is denoted by the purple line. We plotted in orange the average normalised control KS distance, that is, the KS distance between (1) some randomly generated data from a truncated normal distribution of parameters the estimated ones (in green and red) and (2) the theoretical normal truncated distribution of the same parameters. Explanations about the control distance are available in the text. Standard deviations are estimated from 10 samples and the same number of points are used for both the observed and the controlled KS distance. For illustration purpose, the controlled and the observed KS distance have been normalised: both distance are multiplied by a constant enabling them to scale with the other curves.

thresholds, the truncated distribution is indeed a mixture of two gaussian distributions (e.g. the edge and the non edge ones). As the threshold decreases, the proportion of edge weights decreases and the thresholded distribution becomes closer to a truncated normal one so the fit improves. However, as the threshold decreases, the number of non edge weights decreases as well. So it becomes harder to infer the parameters of the non edge weight distribution. At some point, the loss of data points becomes detrimental to the prediction and this explains the U-shape of the observed KS distance. This also explains why the estimated parameters ($\hat{\mu}_n$ and $\hat{\sigma}_n$) diverge from their theoretical values for low thresholds.

It is interesting to remark that the observed KS distance starts increasing (as the threshold decreases) when it reaches the control KS distance (in orange). The control KS distance measures the distance between the inferred theoretical truncated normal distribution (of parameters $\hat{\mu}_n$ and $\hat{\sigma}_n$) and from a set of weights generated using this theoretical distribution (and of size the number of weights smaller than the threshold). Hence this control distance is the smallest possible distance and should be seen as a reference. Note that the control KS distance increases as the threshold decreases. This is because as the threshold decreases fewer data points are available. A match between the observed KS distance and the control KS distance indicates that the observed truncated distribution is as far away from the theoretical normal truncated distribution as a randomly generated set of weights from this distribution would be. This suggests hence that the observed truncated distribution is only made of non edge weights and that it can thus be used to infer the parameters of the non edge weight distribution. This is confirmed by the fact that the observed KS distance reaches the control KS distance when the inferred parameters ($\hat{\mu}_n$ and $\hat{\sigma}_n$) closely approach their actual values.

Figure 4.3, illustrates the second part of the inference process. The predicted proportion of edge weights is plotted in orange and its smooth version in red. First, one should observe that the edge probability starts increasing with the probability density function (p.d.f) of the edge distribution (purple). Second, the edge probability reaches 0.5 when the edge p.d.f (purple) and the non edge p.d.f (cyan) cross. Third, the edge probability reaches one when the non edge p.d.f reaches 0. All of these observations provide evidence of the effectiveness of the proposed method on this simple example. In fact we observe that the predicted number of edges (970) is very close to the exact one (1000) and that

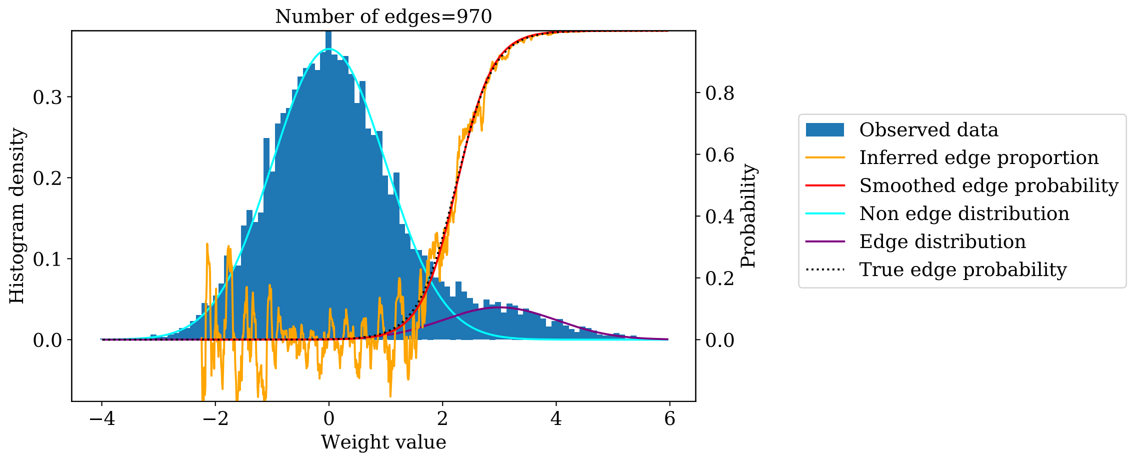


Figure 4.3: Output of the second methodological step. The observed weights are plotted in blue. We also plotted in purple and cyan the weighted theoretical probability distribution functions of edge and non edge weights. The estimated proportions as defined in equation 4.3 are plotted in orange and their smooth version in red. Finally the true edge probability, i.e. the probability that a weight accounts for an edge, is plotted in black.

the smoothed edge probability (red) is very close to the true one (dotted black).

4.3.2 Sensitivity Analysis

In order to analyse the sensitivity of the method, ground truth is required and we hence model the edge and non edges distribution via two known distributions. The relationship between those two distributions is an essential factor determining the efficiency of the proposed method. Hence, to control efficiently this relationship, we start by parametrising the two distributions with two normal distributions and analyse the impact of the parametrisation in section 4.3.2. However, the normality of the non edge weights distribution is predicted by theoretical results only under specific conditions that are often only asymptotically true. It is hence essential to study the impact of any deviation from normality. We proceed to such analysis in section 4.3.2.

Normal distributions

Without loss of generality, we can once again assume that the non edge weights are following a $\mathcal{N}(0,1)$ distribution. This set-up enables us to study three key characteristics of the relationship between the two distributions: (1) their statistical distance modeled by the expected value and standard deviation of the edge distribution (μ_e and σ_e); we can expect that the further away they are the easier it should be to estimate the edge probability, (2) their size, modeled by the total number of weights involved ($N = N_n + N_E$); we can expect that the more weights there are the easier it should be to estimate the edge probability; and finally (3) their relative size, modeled by the percentage of weights that are associated to each distribution ($\frac{N_n}{N} = 1 - \frac{N_E}{N}$). As this is not mentioned in the assumptions of the method, we should expect no impact. Since there are four parameters that model these characteristics, we cannot explore all the possible set-ups hence we use as a reference the scenario used in section 4.3.1 and change one parameter while keeping all other constants. To measure the goodness of the estimation of the edge probability we use as a proxy the comparison between the total number of edges predicted and the theoretical one. This is done for two reasons. First, it is easily interpretable and it is probably the first characteristic one would look at. Second, since the shape of the smoothed edge probability is predefined, the freedom of the smoothed edge probability function is very limited; it is described by just two parameters. Hence an incorrect predicted number of edges is easy to understand: a too large number of predicted edges indicates that the predicted edge probability starts increasing too early. The reciprocal proposition is also true. We plot the results on Figure 4.4 and 4.5.

Figure 4.4 illustrates the impact of the statistical distance between the two distributions. It should first be observed that the predicted number of edges converges to the theoretical one as the distance between the distributions increases, i.e. as the expected value of the edge distribution increases or as its standard deviation decreases. This provides evidence for the correct behaviour of the method under "easy" conditions. As conditions harden, e.g. as the overlap between the distributions becomes more important, we observe that the predicted number of edges decrease and becomes significantly smaller than the theoretical one. This suggests that the method is biased and tends to underestimate edge density. This is most likely a consequence of the assumption made in Equation 4.1. We indeed assumed that the edge density was negligible on the left side of the threshold. We discuss potential improvements in section 4.4. Figure 4.5 analyses the effect of the sample size and of the proportion of edge weights within the distribution. As the total number of weights increases, the statistical uncertainty decreases and the predicted number of edges converges to the theoretical one. Second, as the statistical uncertainty increases, the proposed method tend to underestimate edge density. Finally the analysis of the lower panel

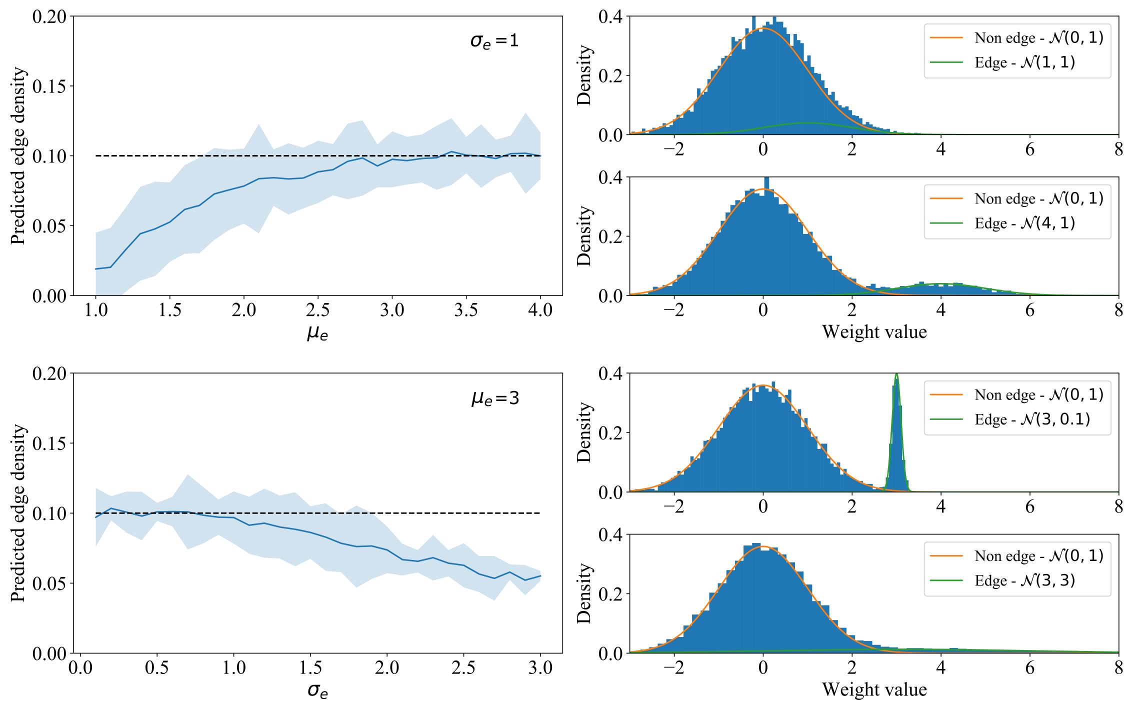


Figure 4.4: Evolution of the predicted edge density as a function of the edge weight expected value while keeping the edge weight standard deviation constant (top) and as a function of the edge weight standard deviation while keeping the edge weight expected value constant (bottom). The true edge density is denoted by the black dotted line. The average value of the predicted edge density and its standard deviation are computed over 100 trials. Two extreme cases are illustrated on the right hand side of each figure where a sample distribution of the weights is plotted in blue and where the theoretical distribution of the non edge weights and of the edge weights are plotted in orange and green, respectively. The aggregated distribution is made of 10,000 weights and the edge weight distribution accounts for 10% of it.

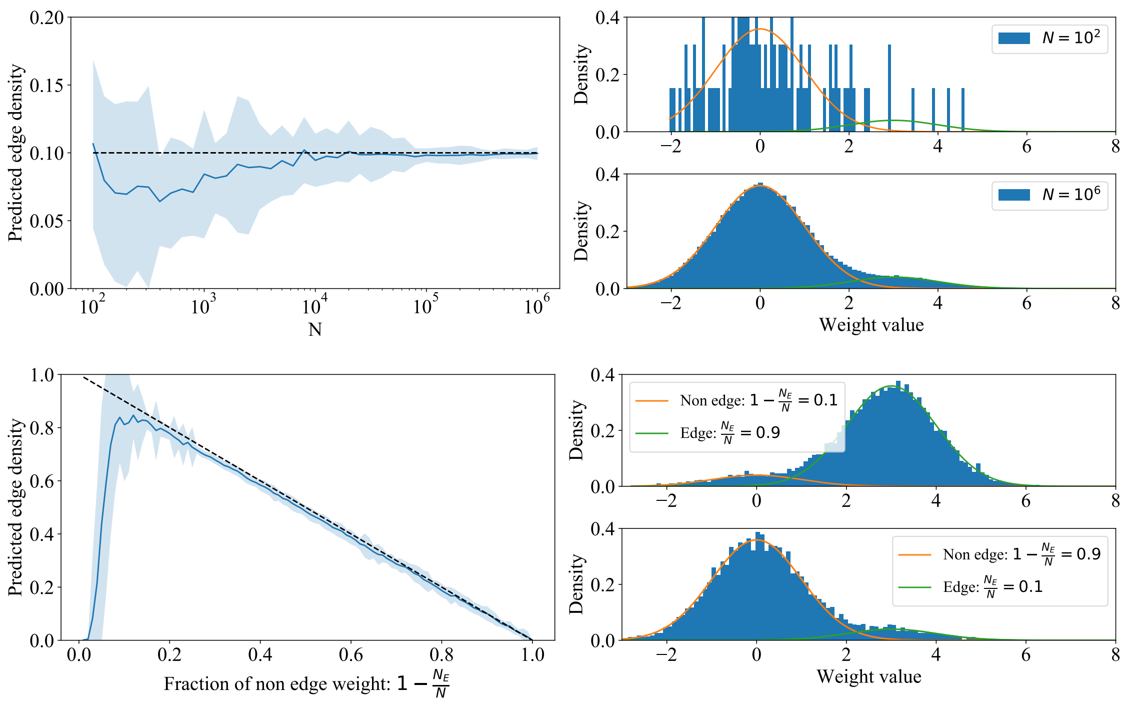


Figure 4.5: Evolution of the predicted edge density as a function of total number of weights used (top) and as a function of proportion of non edge weights (bottom). The true edge density is denoted by the black dotted line. The average value of the predicted edge density and its standard deviation are computed over 100 trials. Two extreme cases are illustrated on the right hand side of each figure where a sample distribution of the weights is plotted in blue and where the theoretical distribution of the non edge weights and of the edge weights are plotted in orange and green, respectively. They are both modeled by a normal distribution of expected value and standard deviation (0,1) and (3,1), respectively. On the top panel, the fraction of edge weight is equal to 10%. In the lower panel, the aggregated distribution is made of 10,000 weights.

reveals that for most edge weight proportions the performance of the method is independent of the proportion of edge weights among the weight distribution. However, for large edge weight proportion, this is not true; the predicted edge density drops rapidly. Note, nevertheless, that high edge density scenarios are characterised by the presence of a left side long tail (see the third right panel of Figure 4.5). Hence, since such tail suggests the presence of a high edge density, such results can hence quickly be a posteriori checked. A left side long tail and of a low predicted edge density should be highly suspicious. We discuss a post hoc analysis in section 4.4.

Non-normal distributions

To quantify the importance of the distributions normality, we make the distributions progressively less normal. The skewness and the kurtosis of a distributions are two widely used statistics that respectively measure the asymmetry of the probability distribution of a real-valued random variable about its mean, and the propensity of a distribution to exhibit outliers. They can hence be used to measure deviation from normality along those two dimensions. Note that two distributions can possibly be modified: the edge and the non edge ones. The analysis made in the previous subsection suggested that the overlap between the edge and non edge distributions was crucially important. Hence to analyse the impact of a deviation from normality, we fix the overlap between these two distributions. Remark that we made no assumption about the edge weights distribution. Hence, given a fix overlap between distributions, we expect changes of the edge weights distribution shape to have no impact onto the prediction. On the contrary, we assumed that the non-edge weight distribution was normally distributed. Hence, given a fix overlap between distributions, we expect deviation from normality of the non edge distribution to negatively impact the performance of the method. Using the same approach as in the previous section and the same reference scenario, we looked at the individual consequences of introduction of skewness and kurtosis. We used the *statsmodel* python package to generate distributions of a given mean, standard deviation, skewness and excess kurtosis and plot the results on Figure 4.6.

An analysis of the left panels of Figure 4.6 reveals that introduction of excess kurtosis or skewness within the edge distribution has little impact. This provides strong evidence for the fact that, given a fix overlap between the two distributions, the proposed method

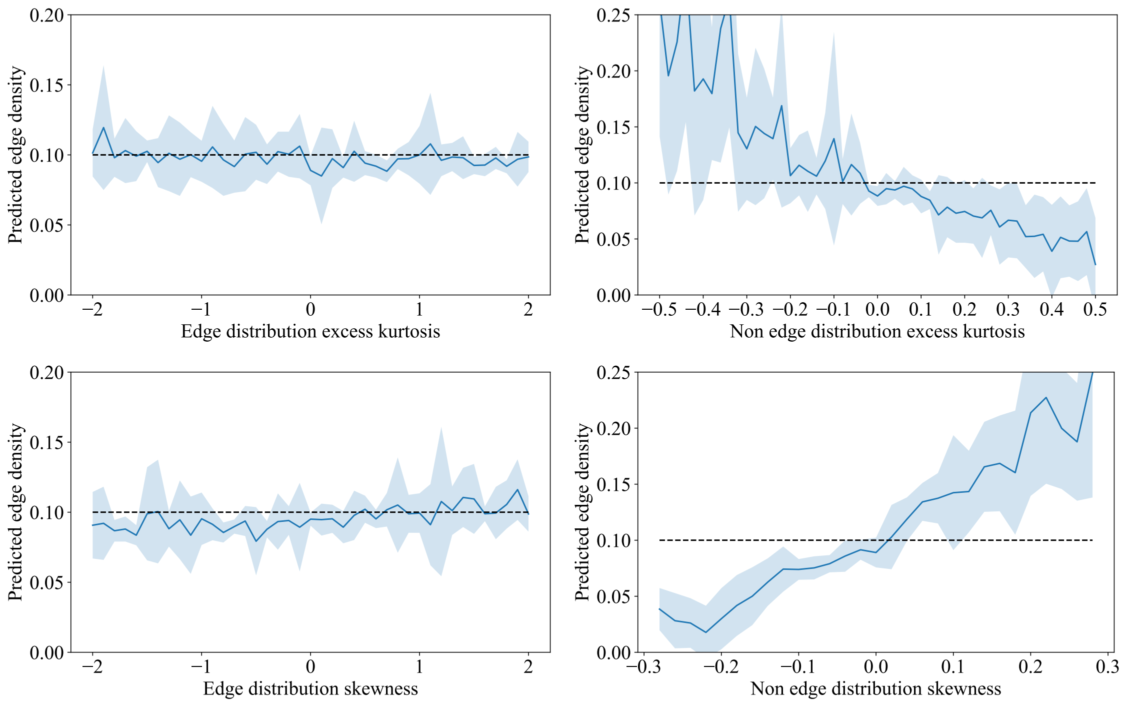


Figure 4.6: Evolution of the predicted edge density as a function of introduction of excess kurtosis (top) and as a function of skewness (bottom) of the edge distribution (left) and the non-edge distribution (right). The true edge density is denoted by the black dotted line. The average value of the predicted edge density and its standard deviation are computed over 100 trials. In absence of skewness or excess kurtosis, 90,000 non-edge weights and 10,000 edge weights are randomly generated from normal distributions of parameters (0,1) and (3,1).

is actually independent of the edge weights distribution. Note that this is critical for the applicability of the method to real world scenarios. Indeed, since interactions between coupled elements are in general unknown, it is not possible to build expectation about the shape of the edge weight distribution. An analysis of the right panels of Figure 4.6, shows however that the proposed method is very sensitive of the degree of normality of the non-edge weights distribution. Observe first that as the skewness of the distribution increases, the number of predicted edges increases. This can be explained. If the non-edge weights distribution is positively skewed, it exhibits a long tail for positive values. Hence since the mean of this distribution is equal to 0, the slope of its p.d.f. is steeper for negative value than that of a normal distribution. Hence the fitted normal distribution is likely to be parametrised with a lower expected value and a lower standard deviation than expected. This results in predicting too few non edge weights. Likewise, if the non edge weights distribution is negatively skewed, the slope of its p.d.f. is less steep for negative value, hence the maximum likelihood approach tend to predict a non-edge distribution with larger expected value and standard deviation. This also results in predicting too few edges. This behaviour also explains why the proposed method is sensitive to the introduction of excess kurtosis: a negative (positive) excess kurtosis increases (decreases) the slope of its p.d.f. for negative values. It hence leads to the prediction of too many (too few) edges. This sensitivity is to be expected since this method is based on the normality of the non-edge weights distribution. Nevertheless, one should observe that the accuracy decrease is gradual. This suggests that the proposed method can adapt to distributions that are slightly non-normal.

4.3.3 Comparison to false discovery rate procedure

Approaches based on false discovery rate (FDR) procedures, such as the Benjamini-Hochberg one [11], predefine a rate, q, and select a threshold t_{FDR} that, given correct p-value associated to every weights, guarantees that the obtain false positive rate will be smaller than q. Given the ordered p-values, $p_1 \leqslant p_2 \leqslant ... \leqslant p_N$, the threshold $t_{FDR}(q)$ is given by:

$$t_{FDR}(q) = \max\left(\frac{q \times k}{N} \mid p_k \leqslant \frac{q \times k}{N},\ k \in \{1, ..., N\}\right). \tag{4.5}$$

Only weights strictly greater than this threshold are accepted. Our approach returns an edge probability, $p_e(w_i)$, for every weight w_i, so, here, we select the threshold t_{F1} that maximises the predicted F1 score based on those probabilities:

$$\begin{cases} t_{F1} &= \operatorname{argmax}_{t \in \mathbb{R}} \left(\frac{2p(t)s(t)}{p(t)+s(t)} \right) \\ p(t) &= \frac{1}{\sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t)} \sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t) p_e(w_i) \\ s(t) &= \frac{1}{\sum_{i=1}^{N} p_e(w_i)} \sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t) p_e(w_i) \end{cases} \tag{4.6}$$

where $p(t)$ and $s(t)$ are the predicted precision and sensitivity, respectively, if we use a threshold t. To measure the quality of the prediction, we analyse the actual F1 score based on the thresholds calculated with our approach and with a FDR procedure. For a threshold t, the actual F1 score is calculated as:

$$\begin{cases} F1(t) &= \frac{2P(t)S(t)}{P(t)+S(t)} \\ P(t) &= \frac{1}{\sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t)} \sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t) \mathbb{1}(w_i \in W_E) \\ S(t) &= \frac{1}{N_e} \sum_{i=1}^{N} \mathbb{1}(w_i \geqslant t) \mathbb{1}(w_i \in W_E) \end{cases} \tag{4.7}$$

where W_E is the set of weights that are actually associated with edges. To evaluate performances we use the same paradigm as in section 4.3.2, except that this time the reference scenario is slightly different: non-edge weights are following a $\mathcal{N}(0,1)$ distribution, edge weights are following a $\mathcal{N}(2,1)$ distribution, $N = 10,000$ and $\frac{N_e}{N} = 0.1$. We used a different edge-weight distribution in order to avoid trivial results where both approaches estimate very well the edge distribution (F1 scores very close to 1). Note that, in contrast with our procedure, FDR approaches assume that they already know exactly every p-values. Although this is debatable, we argue that this assumption is stronger than the one used by our approach, i.e. non edge weight distribution being normal. Indeed, in order to calculate every p-values we have had to assume that the non edge weight distribution was $\mathcal{N}(0,1)$. Hence in order to overcome this informational discrepancy, we apply our method twice: once normally, i.e. without knowing what are the parameters of the non-edge distributions, and once when knowing. Rates q are typically taken to be equal to 10% [89], so we analyse results for $q \in \{5\%, 10\%, 20\%\}$. Finally to get a reference level, we also plot the maximal possible F1 score, i.e. $\max_{t\in\mathbb{R}}(F1(t))$. Results are shown on Figure 4.7.

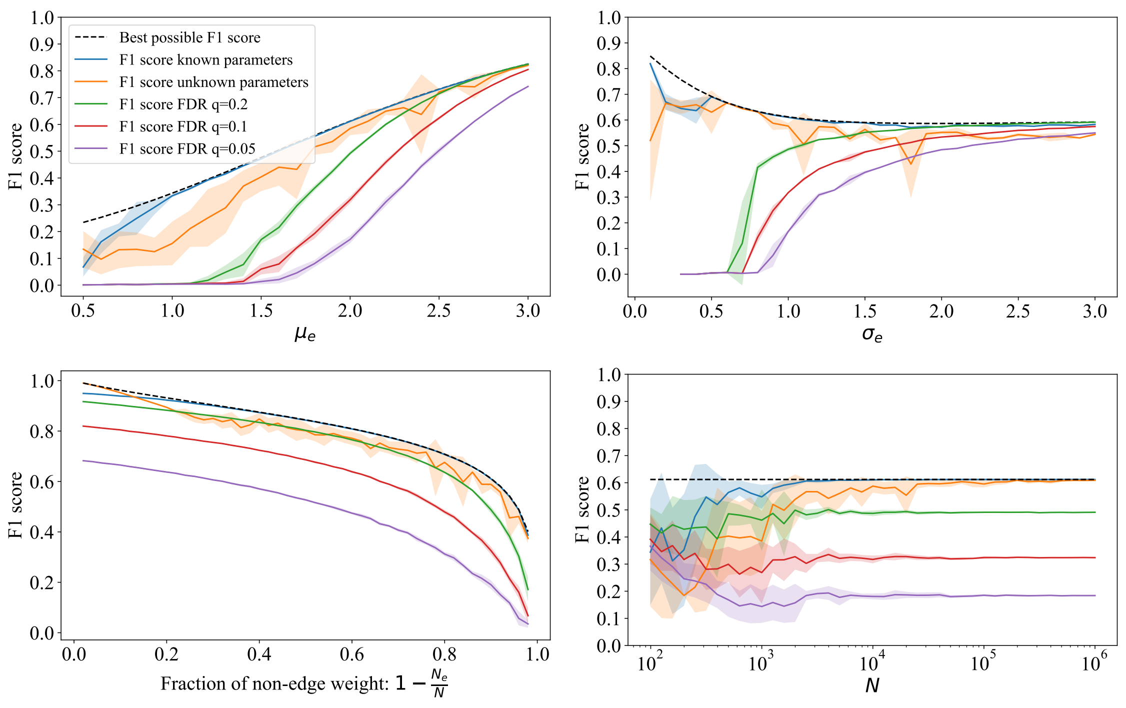


Figure 4.7: Evolution of the F1 score, as calculated in equation 4.7, as a function of the parameters of the edge distribution: μ_e (top left) and σ_e (top right); as a function of the fraction of non edge weight (bottom left) and as a function of the total number of weights (bottom right). Standard deviations are calculated over 10 trials.

Figure 4.7 illustrates the performance of the different approaches. The are a few points to make. First, if the non edge distribution parameters are known, our approach performs extremely well. It is indeed very often matching the best F1 score. This shows that the main difficulty is to get the parameters. Second, there are scenarios for which, whatever the rate q chosen, FDR approaches predict no edges at all. As illustrated on the two top panels of Figure 4.7, this happens when the overlap between the edge and the non edge distributions is important. This is a consequence of the guarantee of a maximum false positive rate. For instance, although the two distributions are manually distinguishable when $\sigma_e = 0.1$, FDR procedure predict then no edges at all. Third, even when our approach is oblivious of the non edge distribution parameters, it performs better than every FDR procedure listed here. This is particularly true when the prediction is harder to make, e.g. for low maximal attainable F1 scores. Fourth, depending on the rate chosen q, the quality of the FDR prediction varies a lot. Since there is no standard way of choosing such a rate, this illustrates the arbitrariness of FDR procedures. Finally, while our two approaches converge to the best possible F1 score as the sample size increases, we observe that false discovery rate approach converges to a value that is a function of the rate q.

Together these observations show that the guarantee of a maximal false positive rate, provided by FDR procedures, comes at the cost of a complete absence of knowledge about false negative rates. Hence, since there are no mean of control over this later rate, such procedures cannot be used to infer network with large F1 scores. The quality of the prediction is dependent on a free parameter q. However there exist no guidelines as for how to tune it. As a result, predictions, with respect to F1 scores, tend to be arbitrary. Furthermore, such procedures make assumptions (correct knowledge of every p-values) which are comparable to the ones we made.

4.3.4 Application to real world data

For the purpose of this thesis, we analyse here the possibility of applying the proposed approach to the set of Z-scores obtained with the time-series extracted from the dataset described in chapter 2. The Z-scores are shown on Figure 4.8. To verify whether we can apply our approach, we would need to check whether the non-edge distribution is normally distributed. Unfortunately we cannot throughly perform such a check since we do not a priori know what are the distributions. We can nevertheless make a few remarks. First, we observe the distribution is heavily skewed on the positive side. This suggests that the edges are making up for this heavy tail. Second, we observe that the distribution is peaking approximately at 0. Hence, as it is reasonable to assume that the non-edge weights distribution is symmetric and centred in 0, this leads us to hypothesize that non-edge weights make up for most of the negative weights. However, as illustrated on the left panel of Figure 4.8, the weight distribution observed on the negative size, is not characteristic of a normal distribution. Third, there are many weights that are simply too negative to have been generated by a $\mathcal{N}(0,1)$ distribution (the expected distribution). Hence, this strongly suggests that the non-edge distribution is not normal. This besides is not really surprising since we know that Z-scores are $\mathcal{N}(0,1)$ for solely independent Bernoulli processes. As a result the application of the described procedure here leads to the prediction of no edge. The most likely explanation lays in the fact that the observed time-series of events cannot be modeled by Bernoulli processes.

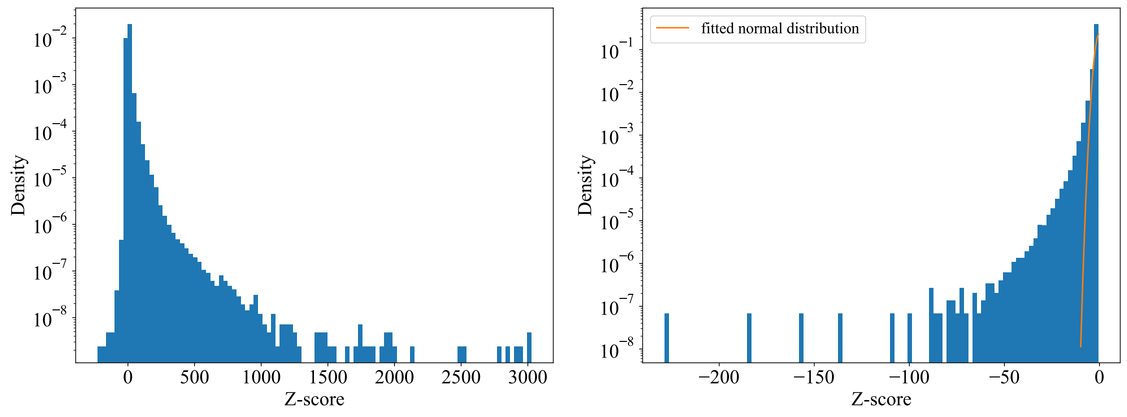


Figure 4.8: Illustration of the distribution of the Z-scores from every possible pair of nodes for the dataset described in chapter 2 (left) Zoom on the part of distribution located on the left side of its median and evolution of the probability distribution function taken from a normal distribution that best fits this part of the distribution (right).

4.4 Discussion

In this chapter, we have developed a statistically principled method that helps one interpreting the weights distribution resulting from the application of a coupling measure to every pair of nodes. Assuming that the weights of non-interacting pair of nodes follow a normal distribution of unknown parameters, we proposed a procedure to find those parameters, the size of the distribution and the probability that each weight accounts for an interacting pair of nodes. To get a thorough idea of the strengths and weaknesses of the method, we started by analysing the accuracy of our approach based on its ability to predict a correct number of edges. We showed that although our approach was generally well working, it was negatively impacted by the presence of an important overlap between the edge and non-edge distributions and that it was very dependent on the normality of the non-edge distribution. Then based on their ability to reconstruct a network as measured by a F1 score, we compared our procedure to FDR approaches. The results showed that in general our approach was reconstructing networks more accurately and that it was less impacted by the presence of an overlap between the edges and non-edges distribution.

Currently, statistically most principled network inference methods [89, 25] select a threshold purely on the ground of FDR procedures [11]. Given a desired false discovery rate q, these procedures provide a minimal threshold value that guarantees a maximal false positive rate. While putting an upper bound on the number of type 1 error may be particularly important when considering very sensitive scenario, e.g. cancer treatment, we argue that this is not the uniquely most important parameter when inferring networks. On the contrary, we believe that network inferences would greatly benefit from approaches that consider both the false positive and false negative rates. Our results showed that our approach was derived to meet such a need and that it was generally more efficient than FDR approaches at recovering thresholds that enable to choose networks with the highest F1 scores. Furthermore contrary to FDR procedures, the applicability of our method is not limited to thresholding and actually suggests a way away from thresholding. Our procedure outputs a function that associates to each weight the probability that it accounts for an edge. Hence, it could also be used to recover weighted networks where each weight could, for instance, measure the probability that it is actually representing an edge. This is an interesting direction because analysis of weighted networks are becoming more popular [130] and can only be more robust since the information carried by each weight has not been binarised.

This method suffers however from three important limitations that require future work. Firstly, it tends to systematically under-estimate the number of edges if the overlap between the edge distribution and the non-edge distribution is important. To address this problem, a possibility is to exploit a source of information that has not been fully exploited. Indeed, once the probability that each weight is an edge has been retrieved, it is possible to extract the distribution of the non-edge weights. Hence a possible approach could try to maximise the normality of this a posteriori retrieved non-edge weight distribution under some constraints. Secondly, our analysis showed that our accuracy is very dependent onto the normality of the non-edge distribution. Any deviation results in a proportional misestimation of the number of edges. Since, the conditions under which the non-edge weights are actually normally distributed are stringent [47, 69], this limits the range of application of the proposed method. Several methods have been developed to empirically estimate the distribution of the coupling measures of non interacting pair of nodes using surrogate data [97]. This hence suggests that methods based on the empirically observed distributions would have most likely a much broader scope of application. Finally, the proposed procedure does not come with confidence level. To address this problem, although this remains to be carefully checked, one possibility would be to analyse the normality of the a posteriori recovered distribution.

We conclude this chapter by stating that, despite the highlighted limitations of the method, the observations we made throughout this chapter, strongly suggest that there is an important informational value within the non-edges distribution that has been under-exploited and that could lead to significant improvement of the state of the art.

4.5 Annex

4.5.1 Parameter estimation of a truncated Gaussian

Consider a truncated gaussian defined over $[a;b]$ of unknown parameters μ and σ we want to estimate. We observe $X = \{x_i, 1 \leqslant i \leqslant n\}$. The log likelihood is:

$$\begin{aligned} L(X;\mu,\sigma) &= log\Big(\prod_{i=1}^{n} \frac{1}{\int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} exp\Big(\frac{-(x-\mu)^2}{2\sigma^2}\Big)dx} \frac{1}{\sqrt{2\pi\sigma^2}} exp\Big(\frac{-(x_i-\mu)^2}{2\sigma^2}\Big)\Big) \\ &= -n \times log\Big(\int_a^b exp\Big(\frac{-(x-\mu)^2}{2\sigma^2}\Big)dx\Big) + \sum_{i=1}^{n} \frac{-(x_i-\mu)^2}{2\sigma^2} \\ &= -n \times log(\Psi_{a,b}(\mu,\sigma)) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} \end{aligned} \tag{4.8}$$

where $\Psi_{a,b}(\mu,\sigma)$ is the integral between a and b of the probability distribution function of a normal distribution of parameters μ and σ. The partial derivatives of $\Psi_{a,b}$ are:

$$\begin{aligned} \frac{\partial \Psi_{a,b}}{\partial \mu}(\mu,\sigma) &= \int_a^b \frac{(x-\mu)}{\sigma^2} exp\Big(\frac{-(x-\mu)^2}{2\sigma^2}\Big)dx \\ \frac{\partial \Psi_{a,b}}{\partial \sigma}(\mu,\sigma) &= \int_a^b \frac{(x-\mu)^2}{\sigma^3} exp\Big(\frac{-(x-\mu)^2}{2\sigma^2}\Big)dx \end{aligned} \tag{4.9}$$

Hence the gradient of the log likelihood is:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial \mu}(X;\mu,\sigma) \\ \frac{\partial L}{\partial \sigma}(X;\mu,\sigma) \end{bmatrix} = \begin{bmatrix} -n\frac{1}{\Psi_{a,b}(\mu,\sigma)}\frac{\partial \Psi_{a,b}}{\partial \mu}(\mu,\sigma) + \sum_{i=1}^{n}\frac{(x_i-\mu)}{\sigma^2} \\ -n\frac{1}{\Psi_{a,b}(\mu,\sigma)}\frac{\partial \Psi_{a,b}}{\partial \sigma}(\mu,\sigma) + \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{\sigma^3} \end{bmatrix} \tag{4.10}$$

It is then possible to maximise the log-likelihood using a gradient ascent algorithm to find the parameters μ and σ that best fit the observation.

Chapter 5

Inferring Functional Connectivity from Time-series of Events in Large Scale Network Deployments

Abstract: *To respond rapidly and accurately to network and service outages, network operators must deal with a large number of events resulting from the interaction of various services operating on complex, heterogeneous and evolving networks. In this chapter, we introduce the concept of functional connectivity as an alternative approach to monitoring those events. Commonly used in the study of brain dynamics, functional connectivity is defined in terms of the presence of statistical dependencies between nodes. Although a number of techniques exist to infer functional connectivity in brain networks, their straightforward application to commercial network deployments is severely challenged by: (a) non-stationarity of the functional connectivity, (b) sparsity of the time-series of events, and (c) absence of an explicit model describing how events propagate through the network or indeed whether they propagate. Thus, in this chapter, we present a novel inference approach whereby two nodes are defined as forming a functional edge if they emit substantially more coincident or short-lagged events than would be expected if they were statistically independent. The output of the method is an undirected weighted graph, where the weight of an edge between two nodes denotes the strength of the statistical dependence between them. We develop a model of time-varying functional connectivity whose parameters are determined by maximising the model's predictive power from one time window to the next. We assess the accuracy, efficiency and scalability of our method on two real datasets of network events spanning multiple months and on synthetic data for which ground truth is available. We compare our method against both a general-purpose time-varying network inference method*

and network management specific causal inference technique and discuss its merits in terms of sensitivity, accuracy and, importantly, scalability.

5.1 Introduction

Swiftly identifying network and service outages to ensure network and service availability in modern, large-scale networks is crucial [58]. Network operators continuously collect log data from all devices and running processes that are deemed to be important. Ensuring continuous network and service availability relies on the efficient and effective analysis of collected data so that outages can be quickly identified or predicted before user experience gets disrupted. This is a very challenging task. Networks are large, complex, heterogeneous and evolving. They support diverse services that are widely distributed, and also evolving. The aggregate rate of collected events is commonly high due to the very large number of monitored devices and services; typical large-scale network deployment can emit up to 10^6 events per second [151]. However, although the aggregate event rate is large, the rate at which individual devices emit events is extremely low such that correlating emitted events is inherently challenging; this becomes even more cumbersome in the presence of periodical informational events [88]. In addition, the vast majority of collected event data is noise and only a few of them may correlate with actionable incidents. Concurrency across network and services results in collected events whose timestamps may be unreliable in terms of absolute values and ordering, due to misconfiguration or loose synchronisation. This makes workflow-based anomaly detection [177] and concurrent log analysis approaches [12] difficult to apply. Finally, there is no explicit model describing the precise mechanisms responsible for the generation of events in the network when an outage or a software failure occurs. For example, black holes due to routing failures may take seconds or minutes to manifest themselves, whereas application servers will start emitting error events immediately after contacting a failed authentication server. Events may not be emitted at all by a failed or failing device/service or a separate monitoring device may emit failure-related events on behalf of unreachable devices after polling a failed device and devices/services that are known to be attached to it (e.g. a server hosting Docker containers or a ToR switch connecting data centre servers), as discussed in [71]. Note that in the latter case, the event emission pattern and frequency is independent of the underlying structural connectivity and solely depends on the configuration of the external monitoring system.

Root Cause Analysis [94] has therefore been a prominent research area. Network operators commonly employ rule-based analysis where a pre-defined and manually updated list of rules is used to exclude uninteresting log data and make analysis of remaining events practical. This is a time consuming and error-prone process. Misconfiguration may result in fatal outages which could have been otherwise easily detected or predicted [151]. Kobayashi et al. [88] recently proposed an inference algorithm for mining causality of network events that has been shown to have good performance. However, the algorithm's complexity is cubic in the number of network events. In model traversing techniques one explores progressively the neighbours of each entity emitting an event to identify its source [83] using a formal representation of the network structure.

In this chapter, we seek to offer a radically different take on how network management operators could go about monitoring, responding to, and even predicting, network and service outages. This new perspective relies on the concept of *functional connectivity* within the network. The term functional connectivity was coined in the field of neuroscience and refers to *"an observable phenomenon that can be quantified with measures of statistical dependencies, such as correlations, coherence, or transfer entropy"*. Importantly, such connectivity is not assumed to denote any causal influence (in which case, the terminology used is *effective connectivity*). From a network management viewpoint, a functional connectivity could therefore denote a number of different things. It could, for example, refer to the integrated involvement of a set of network nodes in the provision of a particular service[1] but it could equally represent a set of network nodes that appear to be systematically involved whenever a particular kind of hardware or software failure occurs. Functional connectivity is underpinned by, but distinct from, *structural connectivity*, a description of the actual physical network infrastructure (including end-hosts, switches, routers, firewalls, NAS devices and any other middleboxes present in the network) typically obtained by taking a dump of the customers operations database. Such database is fed by the change management and connectivity discovery systems, and is automatically updated when network links are provisioned/de-provisioned or equipment is configured.

By design, our approach is agnostic to whether an operational meaning can be readily attributed to the inferred connectivity. Rather, it is a data-driven approach that identifies nodes as having statistical dependencies between their activities. Specifically, we measure these statistical dependencies in terms of properties of the distribution of delays between the events emitted by the nodes. We develop an inference method whose output is an undirected weighted graph where the weight of an edge between two nodes denotes the

[1]Services may be realised at different layers by in-network and end-host devices; e.g., a set of routers that form an OSPF area, a set of switches that are part of a spanning tree, or an application deployment that consists of application and database servers, load balancers and a firewall.

strength of the statistical dependency between them. Our method does not rely on event pre-processing and de-duplication, therefore it is very efficient in terms of execution time and memory requirements. In contrast to [88], we take advantage of all events, including purely informational, periodical events, to infer functional connectivity between network nodes even in the absence of failures or service outages. Depending on the structure of the graph that is being produced (for example, if it consists of multiple connected components or features a strong modularity index), the output of our method can be interpreted in terms of one or many functional groups consisting of a number of nodes; and a node may belong to multiple functional groups (e.g. servers running different VMs supporting multiple cloud tenants' services). With our method a network operator is informed at all times about ever-changing service deployments (and the underlying network topology which can also be seen as a functional one at the physical/link or IP layers). We believe this provides a powerful tool for swiftly responding to, and investigating the root causes of recent or imminent failures, based solely on the times of events emitted by devices.

The chapter is organised as follows. In Section 5.2, we describe the proposed methodology. In Section 5.3, we validate the method by applying it to real-world data and quantifying its predictive power. We then benchmark it against two state-of-the-art methods on both real-world data and synthetic data for which ground truth is available. Finally, we provide results regarding scalability. Section 5.4 discusses research related to our work. We conclude by discussing limitations and possible avenues for further work (Section 5.5).

5.2 Functional Connectivity Inference

Measures of statistical dependence typically used in functional connectivity inference include correlations, coherence and transfer entropy. However, their applicability to event times of devices in large-scale network deployments is severely undermined by (a) the sparsity of events at node level and (b) the lack of knowledge as to how precisely the timing of events is being recorded, or indeed whether this timing is artificially induced by how the network management system obtains or records events (e.g., polling might be involved). Instead, in what follows, we introduce a statistic that meets those challenges and for which confidence intervals have been analytically derived, irrespective of either sparsity or duration of the data. Given two point processes X and Y, each emitting a given number of discrete events m and n on a fixed period of time T, this statistic quantifies the likelihood that the observed number of pairs of events (X_i, Y_j) separated by a delay of less or equal than τ could be expected if X and Y were independent. This statistic is the basis of our assessment that two nodes are functionally connected (i.e., that they are statistically dependent). In the following subsection, we describe the statistic and how it is used in a windowed measure of the temporal relationship between the events emitted by two nodes. This measure (referred to as score thereafter) will then be used to build a model of time-varying edge probabilities (which will be described in Section 5.2.2).

5.2.1 Score: estimating pairwise statistical dependence

The data we were provided with (described more fully in the next Section) consisted of sequences of integer event times (Unix time stamps in seconds) for each device in the network. For the purpose of our statistic, each time-series was interpreted as a fixed-length sequence of 0's and 1's where 1's denoted the presence of an event and 0's the absence of an event. Then, for each pair of nodes, we used a simple adaptation of the cross-correlation function to count the number of times their respective events occurred within less than a given lag (delay) δ of one another, making no distinction between positive and negative delays (thus partly addressing the challenge of concurrency). Formally, we calculated

$$S_{T,\delta} = \{(i,j) \in [\![1,T]\!]^2 : |i-j| \leqslant \delta, X_i = Y_j = 1\}. \tag{5.1}$$

To assess the presence of statistical dependence, we compared this quantity with its expected value and standard deviation when X and Y are independent and identically distributed uniform random variables emitting the same number of events n_X and n_Y over the same period of time T. These values were analytically derived in [102] and in Chapter 3 and summarised here:

$$\mathbb{E}(|\widetilde{S}_{T,\delta}|) = p_X p_Y \Big(T - \big(T-\delta)\big)^2\Big) + p_X p_Y (T-\delta) \tag{5.2}$$

and

$$\sigma_\delta^2 = (2\delta+1)p_X p_Y(1-p_X p_Y) + 2\delta(2\delta+1)p_X p_Y(p_Y(1-p_X) + p_X(1-p_Y)) \tag{5.3}$$

where parameters p_X, p_Y are set to their empirical estimators $\frac{n_X}{T}$ and $\frac{n_Y}{T}$ respectively.

The derivation (see section) of a central limit theorem demonstrating convergence of the distribution of this statistic to a normal distribution of known parameters finally enables us to construct the following Z-score:

$$Z_{X,Y}(\delta) = \frac{|S_{T,\delta}| - \mathbb{E}(|\widetilde{S}_{T,\delta}|)}{\sigma_\delta \sqrt{T}} \tag{5.4}$$

quantifying the likelihood of X and Y being functionally connected.

Since calculating cross-correlations over all possible pairwise interactions is computationally intensive when considering a large-scale network, we typically limited ourselves to a maximum delay δ_{max} as specified in Section 5.3.6 and used the average over all lags up to δ_{max} as our final score.

5.2.2 Model of time-varying connectivity

In this section, we describe our approach to translating the scores introduced in Section 5.2.1 into time-varying probabilities of the existence of functional edges. Since scores require estimates of cross-correlations, a fundamental assumption of the method is that of separation of timescales; changes in functional connectivity should occur much slower than the rate at which processes generate events; this is a realistic assumption in the context of computer network management. Changes in the functional connectivity occur when hardware is commissioned / de-commissioned and services are deployed / un-deployed. Even in very dynamic network deployments that support elastic cloud services, changes in the functional connectivity can be safely assumed to take place at timescales that are significantly smaller than the respective event generation rates (a range of time windows will be considered in Section 5.3). Another source of changes are failing devices (e.g., servers, routers). Such failures do happen frequently, especially in large-scale deployments, however, they result in a stream of events (by neighbouring or monitoring devices) and therefore provide information to our method about functional connectivity around the failing node.

We denote by $s_e(t_w)$ the z-score obtained using equation 5.4 at lag δ_{max} for a potential edge $e = (X, Y)$ within the time window of index $t_w \in \{1, .., N_w\}$, where N_w is the number of non overlapping windows used to decompose the recording. A key principle of the proposed methodology is that the score provides the information required to update the estimate of the value of the probability $p_e(t_w - 1)$ of a functional edge existing between these nodes at the previous time window. More precisely, we consider that information is gained about the probability of an edge existing only when both nodes emit events during the time window considered. This is a natural implication of the sparsity constraint. The fact that only one node in a pair emits an event does not necessarily imply that an edge does not exist (or no longer exists). For each pair of nodes and each time window t_w, there are therefore three cases to consider:

1. The score is positive, $s_e(t_w) > 0$, i.e., there were more coincident or short-lagged

events between these two nodes than between randomly picked pairs of nodes with similar levels of activity. This increases confidence about the existence of an edge and therefore the probability $p_e(t_w)$ should increase as some function h_1 of the score.

2. The score is negative, $s_e(t_w) \leqslant 0$, i.e., there were fewer coincident or short-lagged events than expected at random. This lowers confidence about the existence of an edge and therefore the probability $p_e(t_w)$ should decrease as some function h_2 of the score.

3. At least one of the node does not emit events: This scenario does not provide any information and the probability should remain unchanged.

This leads to the following model formulation:

$$p_e(t_w+1) = \begin{cases} \Big(1-(1-p_e(t_w))\times(1-h_1(s_e(t_w)))\Big) & \text{if } s_e(t_w) > 0, \\ p_e(t_w)\times(1-h_2(s_e(t_w))) & \text{if } s_e(t_w) \leqslant 0, \\ p_e(t_w) & \text{if no information,} \end{cases} \tag{5.5}$$

If h_1 and h_2 are continuous, monotonically increasing and decreasing, respectively, functions of the score with output in $[0;1]$, this formulation ensures that $p_e(t_w)$ remains in $[0;1]$. In our implementation, h_1 and h_2 are simple sigmoid functions, each involving a single free parameter (referred to as α and β thereafter). Other formulations are possible but do not affect the principle of the method, provided they are differentiable in their parameter(s). Since changes in functional connectivity from one window to the other are assumed to be small, we formulate the problem of determining the two free parameters as one of minimising the error of a binary classifier predicting the sign of the score at time t_w given the edge probability at time t_w-1. In other words, if the edge probability at time t_w-1 is greater than 0.5 and both nodes emit events in time window t_w, we expect the score at time t_w to be positive. Conversely, if the edge probability at time t_w-1 is less than 0.5 and both nodes emit events in time window t_w, we expect the score at time t_w to be negative. Our error criterion is formally defined as:

$$E = \frac{1}{2}\sum_{t_w=1}^{N_w}\left(\sum_{s_e(t_w)\leqslant 0 \text{ and } p_e(t_w-1)\geqslant 0.5}(p_e(t_w-1)-th) \quad + \sum_{s_e(t_w)>0 \text{ and } p_e(t_w-1)<0.5}(0.5-p_e(t_w-1))\right). \tag{5.6}$$

It penalises misclassifications, namely $p_e(t_w) \geqslant 0.5$ and $s_e(t_w) \leqslant 0$, or $p_e(t_w) > 0.5$ and $s_e(t_w < 0))$, with a cost proportional to the difference between edge probability and 0.5. As a sum of edge probabilities that are differentiable functions of the parameters, a simple gradient descent can be used to determine the values of the free parameters. Time-varying edge probabilities can then be calculated for all pairs using the update equation (5.5). Because we do not want pair of nodes that never emit events within the same window to have a non zero edge probability, every edge probabilities were initialized to 0.

5.3 Experimental results

5.3.1 Description of the datasets

Real Datasets

This chapter is based on datasets of network events and underlying physical network topologies from two different organisations[2]. The first (**Dataset 1** thereafter) consists of network events and the underlying physical topology from a large retail bank. The infrastructure supports both central and distributed operations in remote sites. The network consists of a core of meshed routers, distribution switches and the supported application and infrastructure servers. The network supports both hub, hub to spoke and intra-spoke operations for financial transactions, and the supporting back office systems. Network events span a duration of 54 days (in period 5/2018 to 06/2018). Structural network topologies were obtained in the middle and at the end of the record. Overall 13,428 different nodes emitted 3,000,418 events leading to a mean value of 4.14 events per node per day. Just 1% of the nodes emitted 65% of all events whilst only 260 nodes (or 2%) emitted more than one event per hour. The second dataset (**Dataset 2** thereafter) corresponds to a Fortune 500 technology company and comprises a core of meshed backbone routers and a distribution layer of switches. It supports the company's commercial operations, including accounting, human resources, research and development, telephony and sales. Network events span a duration of five months (2/5/2015 to 25/11/2015). Considering only those 10,984 nodes in the giant component that emitted events, there were 2,189,579 events leading to a mean value of 1.36 events per node per day. More than half of the nodes emitted only up to 1 event per month whereas less than 3% of the nodes emitted more than 1 event per day. Only a tiny fraction of the nodes emitted more than 1 event per hour.

Construction of the synthetic data

Since validating a method that infers *functional* connectivity is very challenging because more often than not no ground truth is available (as with our real datasets – but see Section 5.3.2), we designed and generated a synthetic dataset capturing as many properties

[2]These datasets are currently not publicly available due to their commercially sensitive nature.

of the real system as possible. First, to enable sensitivity analysis over a large number of scenarios and parameters, we generated scaled-down versions of the actual structural connectivity, i.e., the actual physical network infrastructure. The generation process was as follows. Initialise a first list L_1 with a node n picked at random from the network. Initialise a second list L_2 containing the neighbours of node n. Then, until L_1 reaches the desired size, repeat: choose a member of L_2 at random with probability proportional to its number of neighbours in L_1; add to L_1; update L_2. The resultant graph is the subgraph induced by the vertices in L_1. This process guarantees that the graph generated is connected. We confirmed that this simple process approximately preserved key features of the true topology, specifically, the degree distribution and the distributions of local clustering, local assortativity and betweenness centrality. This is illustrated qualitatively by the top 4 rows of Figure 5.1 and quantitatively by the bottom row, using Jensen-Shannon Divergence (JSD) when varying the size of the synthetic data from 100 to $10,000$ nodes. We note that whilst there are a number of methods able to generate graphs with a prescribed degree distribution (as well as a limited number of other features), we are not aware of any network generative mechanism preserving the above set of properties (whether of the same size or otherwise).

Next, we defined two classes of event-emitting processes. The first class involves functional connectivity in so far as the events are produced as in the four types of failure scenarios identified in [71]. Each scenario involved a different temporal pattern of events. To model *container failures*, in which a server failure leads to events being associated to the contained virtual machines (following probing by the management system), we organised neighbours of high-degree (10+) nodes into a number of disjoint functional groups, each node modelling a different virtual machine. For each container failure, one functional group was chosen at random and events were emitted on behalf of all nodes of that functional group approximately 30s after the failure, with some jitter allowing for bursts spanning approximately 5 seconds. In *intelligent polling failure* scenarios, a similar setup was used except that the events occurred approximately once per minute on a round-robin basis, up to some specified duration. To simulate *flapping interface* failures, we randomly picked two nodes among those nodes with the highest betweenness centrality and constructed a functional group out of all nodes located on the shortest path in the structural connectivity between them. All nodes in the group emitted events at a rate of one every few seconds up to multiple per second for a specific duration. Finally, *service failures*, in which events are generated by dependent applications, involved groups of randomly picked low-degree

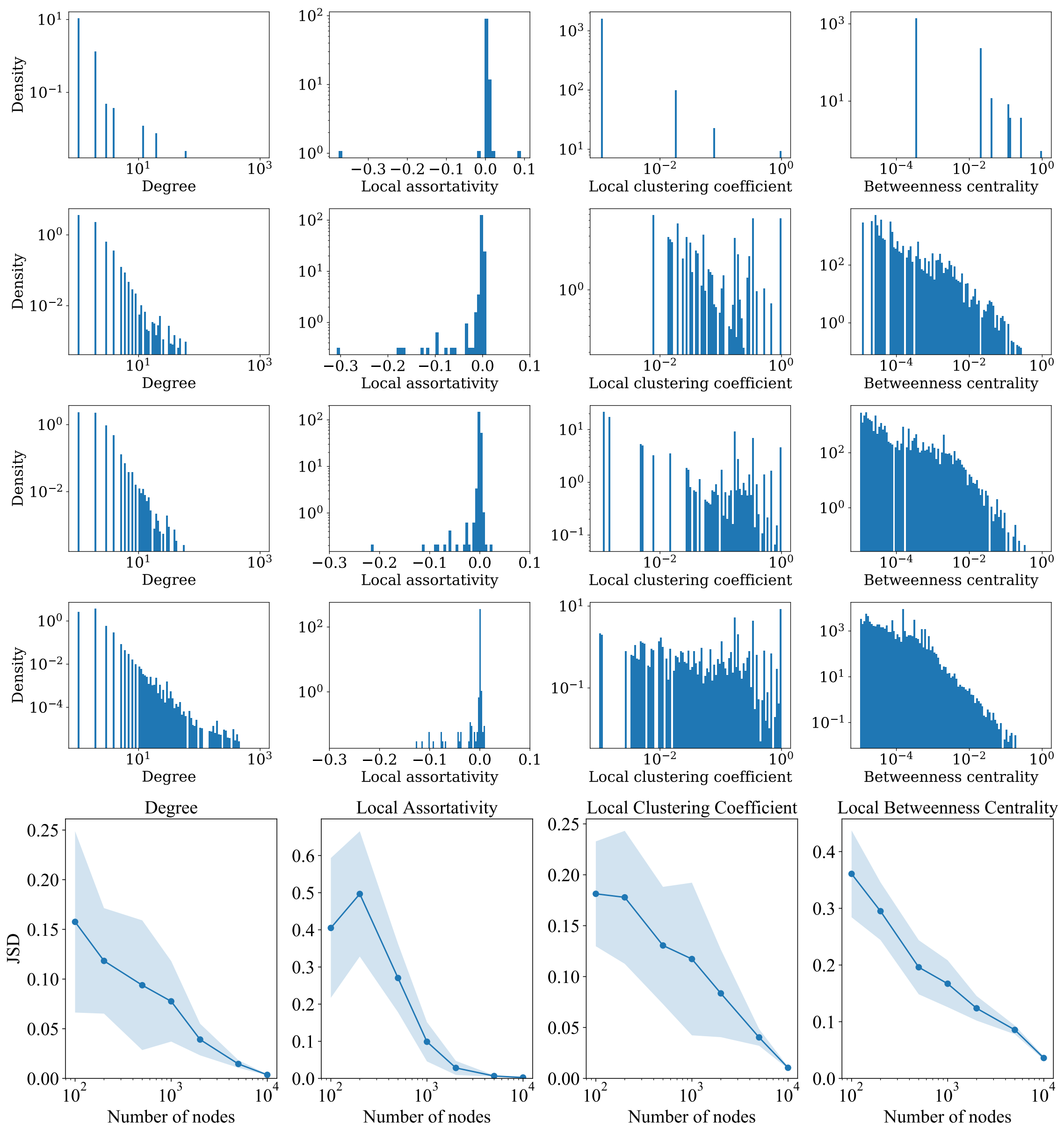


Figure 5.1: (Top 4 rows) Distributions of key network metrics (degree, betweenness centrality, local clustering, local assortativity) for the giant component of the actual network (Dataset 1) (fourth row) and various sizes of synthetic structural connectivities generated using the process described in Section 5.3.1: N=100 (first row), N=500 (second row) and N=1000 (third row). Unsurprisingly, differences in distributions decrease as the size of the synthetic network grows (bottom row). Nevertheless, there is reasonable agreement for networks as small as 100 nodes which were needed to accommodate comparison with state of the art techniques in Sections 5.3.4 and 5.3.5.

nodes, i.e., purposefully not linked to any feature of the underlying structural connectivity. Here, failures consisted of bursts of events generated over a short time-span of up to a few seconds and occurring at random intervals.
In all four cases, temporal changes in functional connectivity were controlled by a parameter determining the probability of a functional connectivity starting/stopping on a daily basis.

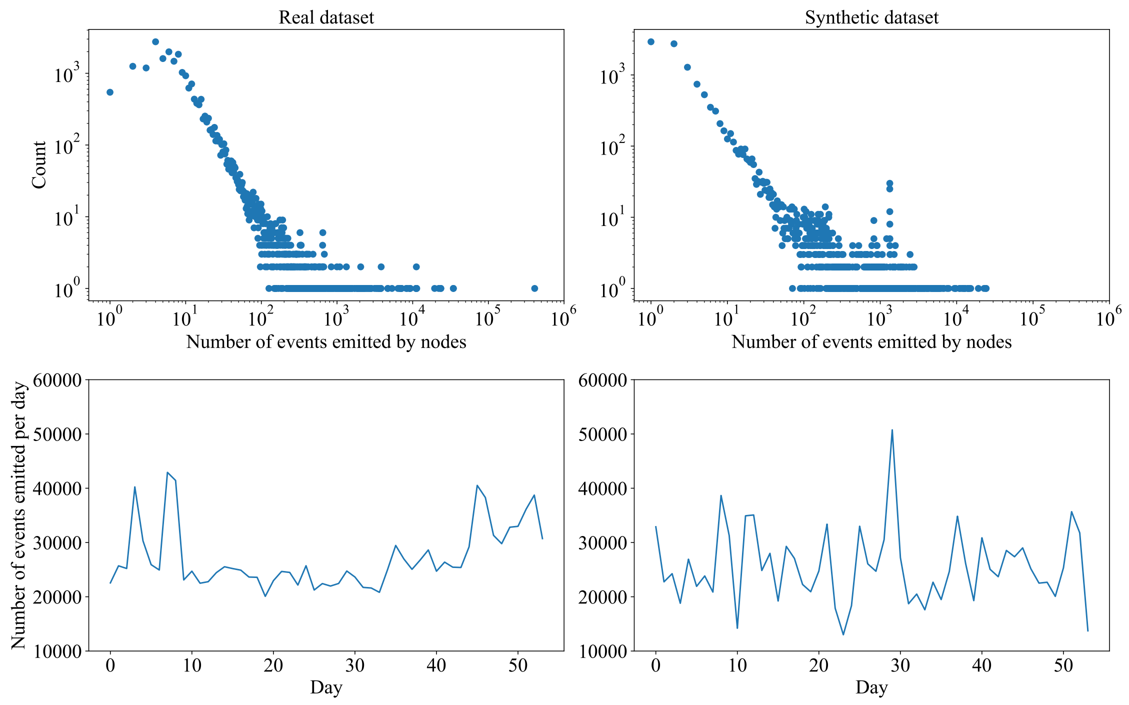


Figure 5.2: Distribution of the number of events per node (top), distribution of the cumulated number of events emitted per day (bottom) for the real (left) and synthetic (right) datasets.

The second class of event-emitting processes produced events occurring at random times on randomly chosen nodes (background noise). The rates of such events for each node were set so that the distribution of the number of events per node (over the entire synthetic dataset) roughly followed that of the real dataset (modelled for simplicity as a power law distribution with exponent $\alpha = 1.8$). Figure 5.2 provides a comparison of both distributions, along with a comparison of the number of events emitted per day. It can be observed that the daily rates for the synthetic data show some burstiness although less pronounced than in the actual dataset.

5.3.2 Validation: Pseudo-ground truth

Acquiring ground truth for evaluating our method is extremely difficult from a practical point of view, as this would require being able to label each device in the network with the different functional connectivities it is a member of, at all times. In the following, we use the type of events a device emits as the proxy for assessing our method in the absence of (absolute) ground truth. The intuition behind this assumption comes from our previous work [103], where we extensively studied Dataset 2 and discovered that only 0.5% of all devices emit events of more than one type and that devices seem to be emitting events that pertain to their functionality in the network. Types refer to the functionality of the running service emitting the event; examples of event types include ‘router’, ‘LANSwitch’, ‘JVM’, ‘Linux’, ‘NetApp’, ‘VMWare’, indicating functionality related to routing and switching, virtualised servers, and storage.

In order to quantify our intuition, we define an event-type distance metric on the types of events emitted by pairs of nodes. Specifically, if devices i and j emit events of types $\{t_1, t_2, ..., t_n\}$ in proportion $\{p^i_1, p^i_2, ..., p^i_n\}$ and $\{p^j_1, p^j_2, ..., p^j_n\}$ such that $\sum_{k=1}^{n} p^i_k = \sum_{k=1}^{n} p^j_k = 1$, then we define their event-type distance as follows:

$$d = \frac{1}{2}\sum_{k=1}^{n}(p^j_k - p^i_k)^2 \tag{5.7}$$

Based on our findings from [103], we expect that nodes belonging to the same functional connectivity will have low values of d, compared to nodes that are not functionally connected. We ran our method with Dataset 2. More specifically, we used 85% of the recording (130 days) to train our method (i.e. optimising the free parameters) and the remaining 15% (21 days) to compute the functional connectivity. To maintain consistency with the method of Kobayashi et al. [88], the window (i.e. the unit of adaptation time, i.e., when probabilities are updated) was set to 1 day. The maximum delay τ_{max} over which cross-correlations were calculated was set to 120s. This is consistent with the values used in the subsequent sections. Figure 5.3 shows the event-type distance d for different values of the threshold on the edge probability. We observe that for all values of the edge probability threshold, the calculated average distance for all pairs of nodes connected in the functional topology output by our method is at least half the average type distance for events emitted by devices not connected in the produced functional connectivity. For comparison purposes, we also provide the distance metric when considering structural links (orange dashed line). It is evident by the average type distance, that physically connected nodes rarely emit events of the same type. We note that, as the threshold value increases, the distance appears to decrease, in line with our intuition; as the criterion for identifying an edge in the functional topology gets stricter, nodes inferred to be functionally connected appear increasingly more likely to emit events of the same type.

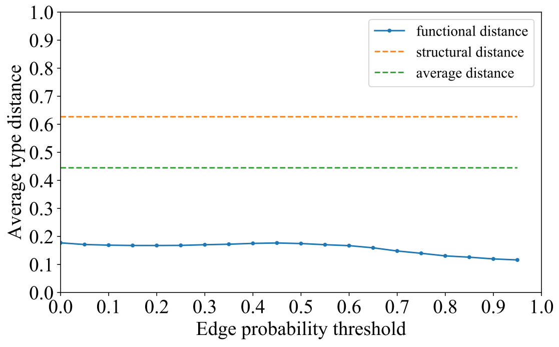


Figure 5.3: Evolution of the average type distance over all connected pairs of device as a function of the probability edge threshold (blue). Average type distance between not functionally connected pair of nodes (dashed green). Average type distance between structurally connected pair of nodes (dashed orange).

Analysis of the functional connectivity. For all sensible values of the edge probability threshold, our method produces a graph that consists of a giant component and a number of smaller connected components (each one consisting of 10 or fewer devices). The distribution of component sizes, along with the number of events emitted by devices belonging to these components, is shown in Figure 5.4.

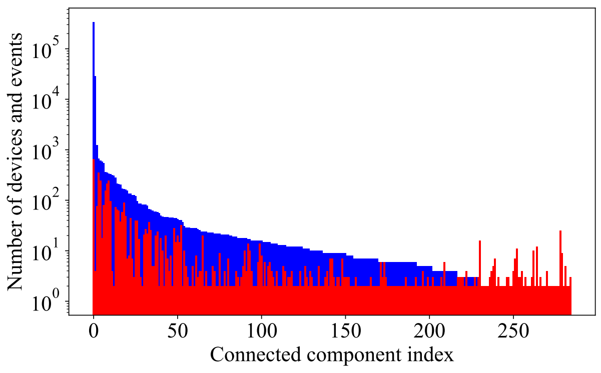


Figure 5.4: Number of devices per connected component (red) and events they emitted over the last 3 weeks of recorded data (blue). Only components of more than one node are displayed. The probability threshold is set to 0.8.

The giant component consists of more than 80% of the network nodes. It encompasses multiple functional connectivities which, as we demonstrate below, are easy to cluster and retrieve. Figure 5.6(left) illustrates the graph that our method produced (with the parameters described above) for a threshold value of 0.8. Each colour identifies a unique event type. Whilst nodes are coloured according to their most frequently emitted event type, edges are coloured according to the types of their end-points if those are identical, black otherwise. Note that more than one clusters of the same colour exist; i.e., the method can identify distinct functional connectivities from the time series of emitted events even when these events are of the same type. For example, two different and independent virtualised server deployments may be emitting the same type of event (e.g., 'VMWare'), even though they belong to separate functional connectivities. The fact that the various clusters identified in the graph appear to be fairly homogeneous in colour suggests that our method is indeed able to identify functional connectivities (provided that event types are an appropriate surrogate of functional connectivity, as suggested by our analysis of the data [103].

To better understand the relationship of functional connectivities and types of emitted events, we used the Louvain community detection algorithm [16] to identify subcomponents in the giant component. We also included the small connected components identified by our method (which we assume to be independent functional connectivities of their own). In Figure 5.5 we illustrate the proportion of the most common type in each functional connectivity, as a function of the edge probability threshold. If a functional connectivity c consists of $|c|$ nodes that emit events of types $\{t_1, t_2, ..., t_n\}$ in proportion $\{p_1^c, p_2^c, ..., p_n^c\}$, the proportion of the most commonly emitted type is given by the weighted sum of the respective proportion for each functional connectivity:

$$prop = \frac{\sum_c |c| \times max_{1 \leqslant i \leqslant n}(p_i^c)}{\sum_c |c|} \tag{5.8}$$

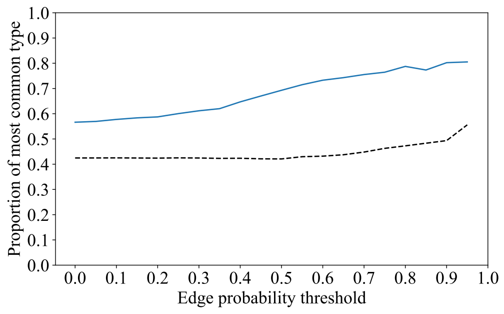
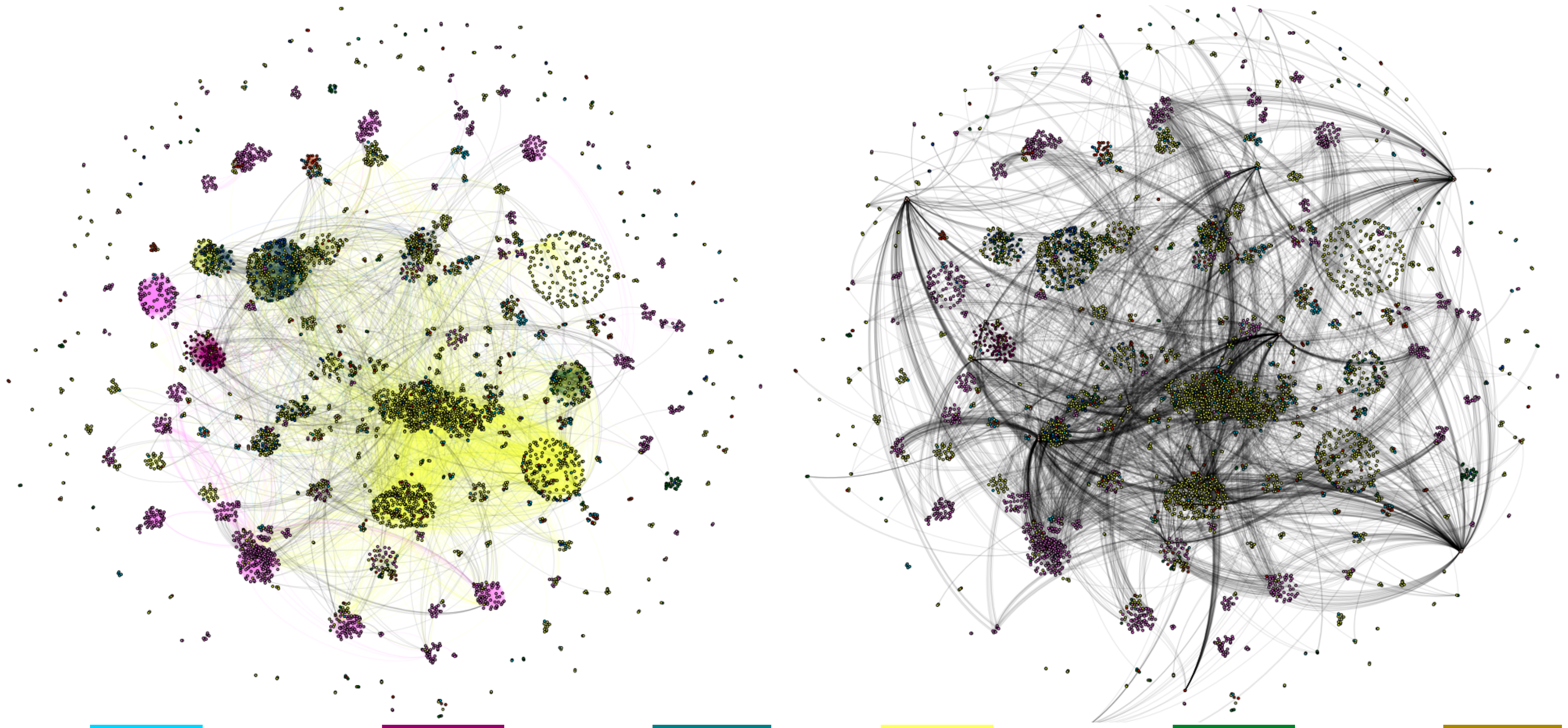


Figure 5.5: Proportion of the most common type in each connectivity (as defined in Equation 5.8) as a function of the edge probability threshold. The black dotted line denotes the proportion of the most common type found across all connected nodes irrespective of community membership.



NT	🟦	JVM	🟪	HPUX	🟦	Linux	🟨	VirtualHost	🟩	InternalHost	🟫
-	🟧	NetApp	🟧	JVM Logs	🟫	VMWare	🟪	CompositeHost	🟦	UCSM	🟩

Figure 5.6: (Left) Functional topology of the graph for a probability threshold of 0.8. (Right) Structural topology using the mapping of the functional connectivity on the left. Detailed, zoomable versions of those figures are available from `https://figshare.com/s/7cbfda9df3222e37710e` and `https://figshare.com/s/9b59a8f1ef882124700e`, respectively. (Bottom) Mapping of colour to type. Nodes are coloured according to their most frequent type and are spatially clustered with nodes they are functionally related to. Hence, functional clusters tend to be made of same type nodes. Edges are coloured if they link two devices of the same type. Hence functional edges (left) tend to connect same type devices more often than structural edges (right). Structural edges tend besides to connect different functional clusters more often than functional edges.

First, we observe that the proportion of the most common type across all connectivities is significantly higher than that when ignoring functional connectivities, confirming that our method does identify groups of nodes that (mostly) emit events of a single type (we remind the reader that event type is not part of the information used to infer connectivity, only event times are). Second, this proportion increases with the threshold, which indicates that the precision of the method increases with the threshold; we extensively investigate the performance of our method with respect to precision and sensitivity in Section 5.3.3.

Comparison to the structural connectivity. We have shown that our method has a high propensity to identify groups of nodes that emit events of the same type. Based on our analysis of the data, we believe these groups to be functional connectivities. Here, we show that knowledge of the event types (which, as a reminder, is not used in our inference method) and that of the structural connectivity would not have permitted to extract said functional connectivities. To illustrate this, we once again plot the inferred functional connectivities of Figure 5.6(left) but replacing the functional edges by the known structural edges (using the same colourings scheme). The result is depicted in Figure 5.6(right).

This analysis reveals that the functional connectivities inferred by our method cannot be predicted based on structural connectivity. Indeed, functionally connected nodes feature virtually no physical connections between them. Further, we observe that most links are black (indeed, 5150 out of 5207 are black), which show that they connect nodes emitting events of different type. Finally, the graph reveals the presence of hubs, i.e., very small (typically less than 5 nodes) but highly dis-assortative functional connectivities originating thick bundles of connections to various other functional connectivities. Careful examination (it will be helpful to the reader to use the zoomable version of the Figure) reveals that these hubs emit events of type 'NetApp', possibly reflecting data storage devices supporting back-end services. This suggests that the method could provide the means to assist with root-cause analysis.

5.3.3 Validation: Predictive power

To further assess the performance of our method, we consider a more operational perspective on the usefulness of the concept of functional connectivity, namely, predictive power, whereby the inference of a functional link between two nodes enables us to make a statement about the likelihood that if events occur at one of the two nodes, events are also likely to occur at the other node. We first consider **Dataset 1** and investigate the predictive power of the method when systematically varying the length of the data over which the

	Condition Positive		Condition Negative	
PCP*	TP	$p_e(t_w) > 0.5$ $s_e(t_w+1) > 0$	FP	$p_e(t_w) > 0.5$ $s_e(t_w+1) \leqslant 0$
PCN*	FN	$p_e(t_w) \leqslant 0.5$ $s_e(t_w+1) > 0$	TN	$p_e(t_w) \leqslant 0.5$ $s_e(t_w+1) \leqslant 0$
NPC*	One node does not emit event at $t_w + 1$			

Table 5.1: Matrix summarising the definitions used to quantify predictive power. *PCP = Predicted Condition Positive, PCN = Predicted Condition Negative, NPC = No Predicted Condition

method is trained (from 2 to 50 days). The testing period (unseen data) consisted of the first 2 days of data after the training period. To maintain consistency with the method of Kobayashi et al., the window (the unit of adaptation time, i.e., when probabilities are updated) was set to 1 day. For this experiment (and all further experiments unless stated otherwise) the threshold (determining whether an edge existed) was set to 0.5. This is fairly arbitrary, and, as we will discuss in Section 5.5, not necessarily helpful. As evidenced by the bottom panel of Figure 5.7, whilst selecting a low threshold does increase sensitivity, the gain is small in comparison to the loss in precision (almost a factor 2 between precisions at thresholds 1 and 0.5). The maximum delay τ_{max} over which cross-correlations were calculated was set to 120s. Again, this value was chosen to facilitate comparison with Kobayashi et al. as it corresponds to 2 bins of 1 minute. To quantify predictive power in the absence of ground truth, we adopted the following definitions (summarised in Table 5.1). An edge is a true positive if the method predicted an edge and there was short-lagged activity across this edge in the testing period such that the score would predict the presence of an edge. An edge is a false positive if the method predicted an edge and there was some short-lagged activity across this edge in the testing period but the score for this activity would not predict the presence of an edge. True and false negatives are defined as the logical counterparts of true and false positives. It is essential to note that these definitions are contingent to short-lagged interaction happening in the testing period. This is because lack of activity across an edge over a period does not provide any information as to the existence of an edge. The edge might exist but not be active over the period. Such property was explicitly included in the construction of the model (see Section 5.2.2).

Figure 5.7(top) shows the evolution of precision and sensitivity as the length of the

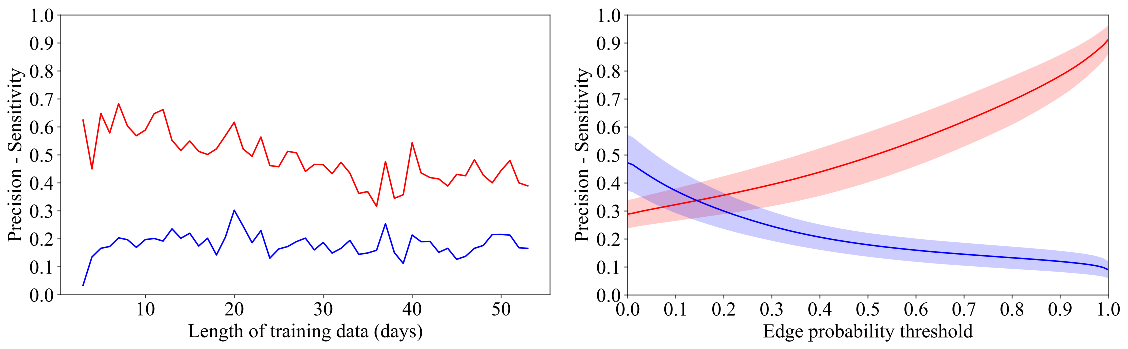


Figure 5.7: (left) Precision (red) and sensitivity (blue) as a function of the length of the training data (single run, full dataset). (right) Average precision (red) and sensitivity (blue) as a function of the choice of edge probability threshold.

training set was varied between 2 and 50 days. Whilst sensitivity remains stable for most length of training data, precision shows a gradual drop as the length of training data increases. This could suggest that the underlying functional topology changed during the record (this will be investigated below). To provide more confidence into the result, we repeated the experiment and averaged performance over 50 subnetworks of 1000 nodes picked at random.

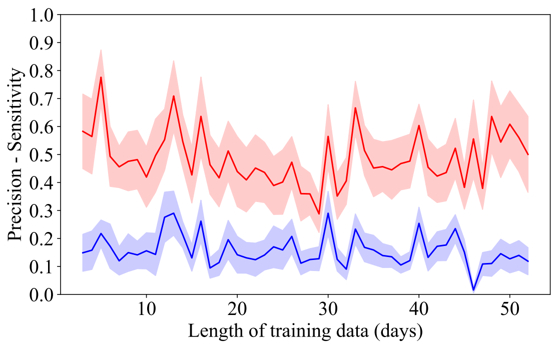


Figure 5.8: Precision (red) and sensitivity (blue) as a function of the length of the training data (averaged over 50 subnetworks of 1000 nodes each).

As shown by Figure 5.8, sensitivity once again showed little sensitivity to the length of training data. Precision was slightly higher, and interestingly, there was less evidence of the decay seen when the full dataset was used. This is a somewhat counter-intuitive observation at first but can be explained in terms of the (limited) ability of a single (small) set of hyper-parameters to model heterogeneity in different components of the underlying functional topology. By considering subnetworks, the amount of per-network heterogeneity is potentially reduced, which may compensate for the potential loss of precision due to a changing underlying connectivity. An interesting operational implication could be that in a highly heterogeneous environment, it might be beneficial to deploy multiple instances of the method (each dealing with specific types of events) rather than one.

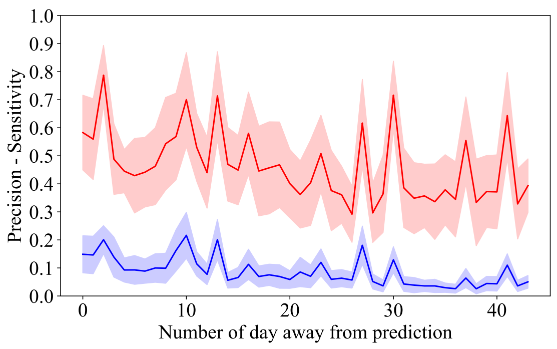


Figure 5.9: Precision (red) and sensitivity (blue) as a function of the time (in days) between training data (first 10 days) and testing data (1 day), averaged over 50 subnetworks of 1000 nodes each.

To shed light on whether the underlying functional topology might have changed (in the absence of ground truth, this is difficult to establish) we analysed the method's predictive power when training was done over 10 days and the testing horizon was systematically varied between 1 and 40 days away from the training data. Figure 5.9 shows some evidence of gradual decline in both precision and sensitivity, suggesting there might have been changes.

To provide some insights into the behaviour of the method in response to changes in the underlying functional topology, we used synthetic data and systematically varied a parameter controlling the amount of changes in the set of functional connectivities involved on each day of the record (specifically, the probability that a functional connectivity starts/stops being active from one day to the other). Fifty networks of 1000 nodes were used, with an average of 20% of the functional connectivities described in Section 5.3.1 active at all times. Figure 5.10 shows that whilst performance remains approximately stable in the absence of changes (the blue line has no slope), suggesting the ability of the method to stabilise its predictions after 10 days, there is a steady drop in performance in the presence of changes, and the drop correlates with the amount of change unsurprisingly.

5.3.4 Comparison with Kobayashi et al. [88]

The method by Kobayashi et al. [88] assumes a direct acyclic graph (DAG) of events corresponding to the causality of events and proceeds in three steps. First they preprocess

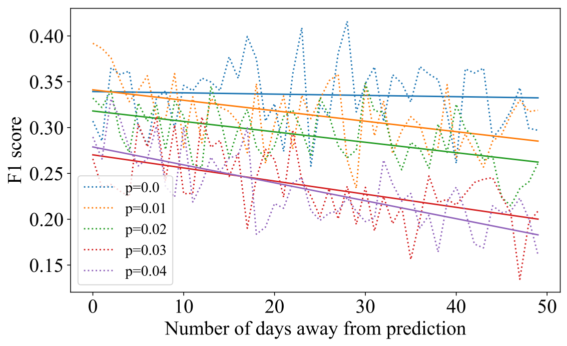


Figure 5.10: F1 score (based on precision and sensitivity as defined in text) as a function of the time (in days) between training data (first 10 days) and testing data (1 day), averaged over 50 subnetworks of 1000 nodes each. Linear regressions are fitted to highlight the trends.

the data and remove events of time series that show strong temporal periodicity. Then, for every pair of nodes (X,Y), they state that an edge is not formed if there exists at least one node Z, such that nodes X and Y are conditionally independent ($P(X, Y|Z) \approx P(X|Z)P(Y|Z)$). Independence is tested using the conditional cross-entropy and the G-square test:

$$G^2 = 2mCE(X, Y|Z),$$

where m is the duration of the recording. Finally, they post-process the data and remove frequently appearing edges to enable the detection of unusually important causality. Our results were obtained using the authors' implementation available at `https://github.com/cpflat/LogCausalAnalysis`.

A comparison between their method and ours is challenging for three main reasons: (1) their method outputs DAGs denoting causal relationships between events based on activity taking place over a day, whereas our method infers an undirected functional topology based on activity taking place over a chosen amount of time; (2) their method requires event descriptors; (3) its greater time complexity (see Section 5.3.6) makes it very impractical to deploy on the kind of large datasets for which our method is designed.

In principle we could apply the same experimental schedule as in Section 5.3.3, however, we found that unlike with our method, the daily networks inferred were changing substantially (e.g., from 23% overlap on consecutive days to 5% over 2 days). The reason for this was found to be the low density of events such that events occurring on the day were not necessarily representative of events occurring on a different day. This means that training the method over 10 days or over the last of these 10 days would yield the same set of DAGs, thus making a comparison with our method unfair. Since only sensitivity is affected by this property, when using real data, we only report precision.

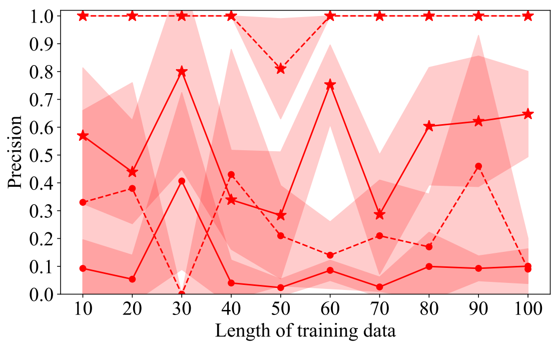


Figure 5.11: Precision using our (star) and Kobayashi's (circle) scoring method for the method of Kobayashi et al. (dashed line) and ours (solid line). Data correspond to mean and standard deviation over the 10 days of unseen data.

Because Kobayashi et al.'s method requires event descriptors, we used **Dataset 2**. Both methods were trained over an increasing number of days and performance was measured over the next 10 days of unseen data. Since Kobayashi et al. predict causal interactions whilst our method returns undirected networks, precision was calculated using both criteria. Figure 5.11 shows that irrespective of the performance measure used, the method by Kobayashi et al. yields significantly higher precision. However, this must be put in context of a huge discrepancy in the number of events being predicted. Whereas our method returned over thousands of functional edges on a daily basis (up to over 100,000 for the longest training periods), the method by Kobayashi et al. only rarely returned more than 10 events per day (2.6%) and often (80%) none at all with an average of 1.2 edges predicted per day. Below, we will use synthetic data to make this case more fully.

The lack of stability in day-to-day inference as well as the low number of predicted events returned by the method of Kobayashi et al. can be attributed to the low density of events per node per day in the dataset (0.5). Experiments (results not shown) revealed that with higher densities, the percentage of overlap between day-to-day predictions increases (up to 30% for 100 events per node per day) albeit far inferior to that of our method (>90% overlap for densities from 0.7 to 100 events per node per day) and at the cost of much longer computations (both our method and that of Kobayashi et al. have a dependence on the number of events to be considered). In what follows, we used synthetic data with a sufficiently high density of events to provide both precision and sensitivity. The testing protocol considered is equivalent to that in Section 5.3.3 but using static networks

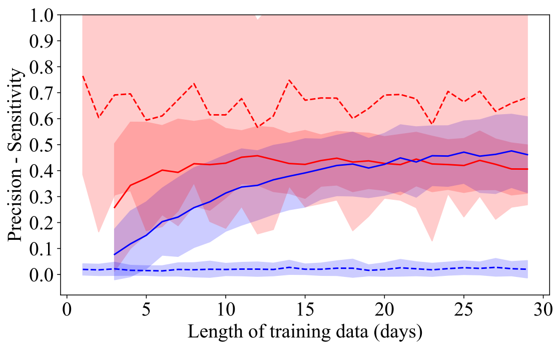


Figure 5.12: Precision (red) and sensitivity (blue) for our method (solid line) and Kobayashi et al.'s method (dashed line). Length of training data is varied between 3 and 29 days. Performance is measured against ground truth on the next day.

of 300 nodes over 30 days (to reduce computational cost). Figure 5.12 shows that whilst the precision of our method is somewhat inferior to that of Kobayashi's, our sensitivity is far superior. Based on our earlier observation that the choice of threshold 0.5 whilst slightly improving sensitivity, dramatically reduces precision, we also calculated precision and sensitivity for a threshold of 0.9. This led to 20% improvement in precision (still less than Kobayashi) and a 33% drop in sensitivity (results not shown).

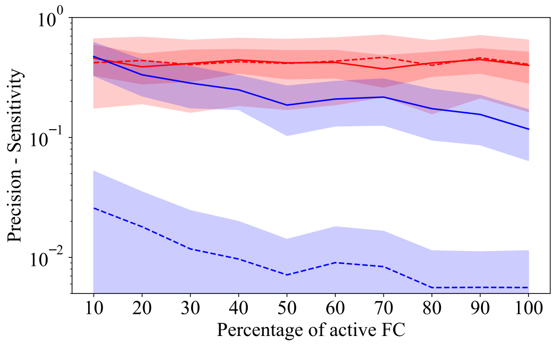


Figure 5.13: Precision (red) and sensitivity (blue) for our method (solid line) and Kobayashi et al.'s method (dashed line) when the percentage of active functional connectivities os varied between 10% and 100%, all other parameters being equal. To make a straightforward comparison possible, our method was configured to return the same precision as that of Kobayashi et al., thus the near perfect overlap between red curves.

It will be noted that we used a very low density of functional edges. As illustrated in Figure 5.13, this is because in the method of Kobayashi et al., the likelihood of an edge existing relies on a p-value derived from conditional cross-entropies. The higher the density of functional edges, the more likely it is to find a node Z such that there is conditional independence between X and Y. There is also a drop in sensitivity for our method. This is because the number of pairs showing significantly higher scores than those picked at random is dropping.

5.3.5 Comparison with Hallac et al. [65]

Hallac et al. [65] extended the graphical Lasso algorithm and developed a method to solve for $\Theta = (\Theta_1, \Theta_2, ..., \Theta_T)$ a set of symmetric positive definite matrices:

$$min_{\Theta \in S^p_{++}} \sum_{i=1}^{T} -l_i(\Theta_i) + \lambda||\Theta_i|| + \beta \sum_{i=2}^{T} \Phi(\Theta_i - \Theta_{i-1}),$$

where T is the number of windows, $l_i(\Theta_i) = n_i(\log \det\Theta_i - Tr(S_i\Theta_i))$ is a function that encourages Θ_i to be close to S_i^{-1} the inverse of the empirical covariance (if S_i is invertible), n_i is the number of observations, $||\Theta_i||$ is the semi-norm of Θ_i, λ is a positive constant that is adjusted to enforce the sparsity of the covariance matrix, $\Phi(\Theta_i - \Theta_{i-1})$ is a convex penalty function minimised at $\Phi(0)$, which encourages similarity between Θ_t and Θ_{t-1} and β is a positive constant determining how strongly correlated neighbouring covariance estimations should be. The connectivity at time t is then simply extracted from the non-zeros values of the inverse of the precision matrix Θ_t. Our results were obtained using the authors' implementation available at `https://github.com/davidhallac/TVGL`.

The challenge of providing a comparison is in setting a suitable criterion for setting the various parameters of Hallac et al.'s method, most critically, number of windows, bin size, choice of penalty function and parameters λ and β. Since the number of windows is an arbitrary choice, we set it to 1 day, as in Kobayashi et al. The penalty function was set to the Laplacian because smooth changes are assumed to take place in the real data. All other parameters were subjected to grid search to maximise the resulting F1 score. In the absence of ground truth, predictive power was assessed in terms of the method's ability to predict an event. This is substantially different from any of the tests used previously but is fair, if not particularly favourable to either method.

The presence of an edge in Hallac et al.'s method was assessed on the basis of non-zero values of the inverse of the Θ matrix it returned. As shown in Figure 5.14, our method outperforms that of Hallac et al. for all configurations considered. In particular, because the networks are so small (size chosen due to the poor time complexity of the method of Hallac et al.), our method can achieve high precision and sensitivity after only 10 days of training data. In a final experiment, we examined whether the density of events (within the limit of what was computationally possible) could explain the poor performance of

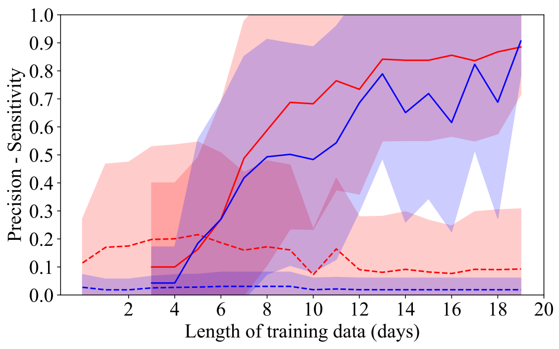


Figure 5.14: Precision (red) and sensitivity (blue) for our method (solid line) and that of Hallac et al. (dashed line) when length of training data is varied between 1 (3 for our method) and 20 days. Performance was calculated as the mean over the 10 days of unseen data and was averaged over 10 networks of 50 nodes each. On average 20% of the functional connectivities were active.

Hallac et al's method. We systematically varied the density of events through multiplying the functional event rates to reduce the sparsity of the binned matrix used by Hallac. As shown in Figure 5.15, whilst the increased density did result in a higher sensitivity for our method, both precision and sensitivity for Hallac et al. remained very low.

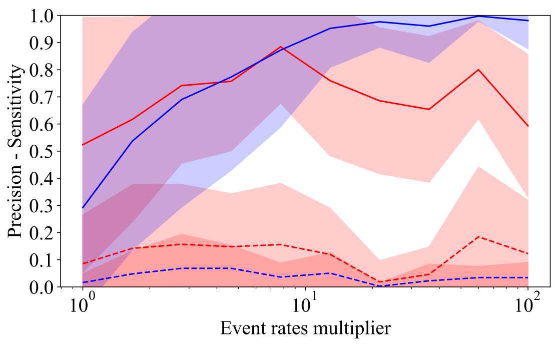


Figure 5.15: Precision (red) and sensitivity (blue) for our method (solid line) and Hallac et al.'s method (dashed line) when the density of events is varied between 1 and 10,000 events per day (for networks of 100 nodes). Performance is measured against ground truth.

5.3.6 Scalability Analysis

A major challenge in this work was the inability of the benchmark methods to handle the size of the real datasets considered here. In this Section, we provide a comparative analysis of how each method scales with network size. For the purpose of this analysis, we ignored any consideration of performance but focused on measuring time complexity when all three methods were set to operate on an as similar setup as possible. Concretely, we used synthetic data and simulated activity over 10 days. Window size was set to 1 day in all three methods. Bin size was 1h for Hallac et al.s' method. Cross-correlations were calculated up to 2 minutes in our method.

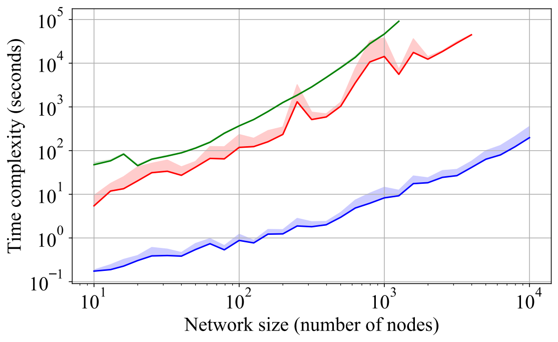


Figure 5.16: Time complexity (in seconds) for our method (blue), Hallac et al. (green) and Kobayashi et al. (red) when network size is varied between 10 and up to 10,000 nodes (depending on methods).

Figure 5.16 demonstrates the clear superiority of our method when the number of nodes in the network is systematically varied, with roughly a factor 10^4 for network sizes of 10^3 nodes. No simulations were carried out for network sizes greater than 10^3 nodes for the benchmark methods due to the excessive time it would have taken to do so.

5.4 Related Work

A number of approaches that are based on traditional data mining techniques have been proposed to extract useful information about the operation of networks and services that could be used for real-time [126, 173] and post-incident analysis [115, 176] and for understanding performance problems[128]. Recently, the use of supervised and unsupervised learning for detecting anomalies in network and service log data has been explored (see [74] for a summary). Workflow construction through processing of collected log data [177, 12] and filtering of unimportant network nodes based on the notion of graph vertex entropy [151] and supervised machine learning [70] have also been proposed. Our method is complementary to the aforementioned algorithms and systems, which can be more efficient and effective when applied to smaller datasets of collected network log data known to have been produced by network devices and servers belonging to the same functional topology, i.e., collectively providing a specific service or core network functionality.

Our work is inspired by research conducted in the context of other complex networks.

In brain networks, even the most advanced forms of imaging cannot provide an accurate or complete description of structural connectivity, see [154] for example. Although link prediction methods are being developed to attempt to extract missing information [95, 98], a more promising approach is to infer connectivity from the temporal evolution of events occurring at node level [24] without assuming any prior knowledge regarding connectivity. In brain neural networks, for example, network inference may rely on spike dynamics [23]. In gene regulation, published methods primarily use gene expression data derived from micro-arrays [75].

A number of methods have been developed to infer connectivity changes in a network by adapting Bayesian network methods [38, 164, 123, 143]. These methods assume a model of event propagation. At their core is the belief that the current state can be predicted, with some probability, based on the previous state or states. In the context of very sparse data, such as in computer network and service deployments, when an event in A may trigger an event in B in only a small percentage of the cases, such an assumption is problematic. Further, and as should be apparent from the failure scenarios we discussed in the context of generating the synthetic data, there can be great heterogeneity in how event times 'follow' from a root cause event. For example, when intelligent polling is used (whereby a monitoring server polls devices that might be affected by a failure elsewhere) can result in event times that are not intrinsic to the network infrastructure itself but rather depend on how the monitoring system is set up to react to the root cause (such information not being available to us). For this reason, methods such as the Markovian model used in temporal exponential graphs [64], various adaptations of the Kalman filter [30, 85] or methods relying on propagation of cascades [124] are unlikely to be as effective as a method that will solely rely on pair-wise information. In fact, the use and adaptation of pair-wise correlations is the basis of many methods that do not assume an event propagation model, e.g., [114, 100, 181]. However, these adaptations typically result in methods that do not scale well to large networks over long recordings.

A final class of methods relies on the estimation of a time-varying covariance matrix to encode the correlation structures at each observation, e.g., [106, 171, 182]. Constraints of sparsity (in the network of interdependencies between the nodes) are enforced by way of lasso penalty. These methods typically do not scale well to large examples, as we showed for Hallac et al. [65].

5.5 Conclusion

Monitoring, responding to, and predicting, failures in a large scale network deployment is a key responsibility of network operators. The sheer amount of data generated by devices makes this task particularly difficult. In this chapter, we sought to present an alternative framework to representing and modelling network events. This framework is based on the concept of functional connectivity first introduced in neuroscience. In contrast to structural connectivity, the underpinning physical infrastructure, functional connectivity represents statistical dependences between the activities of nodes in the network. In this chapter, we specifically focused on statistical dependence based on the amount of short-lagged interactions between them. We presented a new statistic to robustly assess the presence of statistical dependence between two nodes. By deploying this statistic in a windowed-fashion and embedding it into a predictive framework, we were able to develop an inference model of time-varying functional connectivity able to meet three key challenges: (a) non-stationarity of the underlying structural and functional connectivities, (b) sparsity of the time-series of events limiting the effectiveness of classical measures of statistical dependence, and (c) lack of information as to how events follow from root causes.

The fact that a substantial amount of this chapter was focused on the methodological aspects of inferring functional connectivity in large-scale commercial deployments should not detract from its main focus, namely, making the case for functional connectivity as a powerful tool in the arsenal of network operators. Through our various validations effort, we have sought to demonstrate a number of benefits. First, as shown in Section 5.3.2, the inference of functional connectivity provides the kind of insights that might otherwise require intensive processing of the content of the events. We showed that the extracted communities were characterised by (among other things) great homogeneity in the type of events that originating them. Such information can be available to operators but commercial experience (as well as academic papers such as [88]) shows that recovering it typically entails a significant cost because event descriptions can involve templates that change over time, manual entries that might be subject to human error, etc. Further, relying on event descriptions constrains the kind of functional relationships that will be identified to those that will have been anticipated by the operators setting up the system, e.g., services, expected failures. By relying on timestamps only, the concept of functional connectivity is agnostic to the origin of the dependencies. It tells the network operator that nodes that might not have been predicted to be related in any way, actually are, due to unexpected factors. Second, the concept of functional connectivity alleviates the need for accurate network discovery that afflicts most existing commercial systems. Currently fault localisation and root cause analysis depend on an accurate description of the network. This is challenging for two reasons. First, the network is complex, heterogeneous and changing. Second, automated discovery approaches cannot always capture the inherently multiplex nature of large-scale deployments. Instead, although functional dependencies are underpinned by structural connectivity, their identification is not contingent on having full and accurate knowledge of this structural connectivity. This was clearly illustrated in Section 5.3.2, and Figure 5.6 particularly, showing the almost total lack of overlap between functional and structural connectivities. Indeed, we found an almost complete lack of structural links between nodes belonging to a functional connectivity (as identified by community structure membership). Instead, structural edges appear to link functional connectivities, highlighting their mediation but not causal role in the emergence of functional relationships. Thus, at the very least, one can think of functional connectivity as providing added value to current network discovery protocols. Finally, inference of functional connectivity, including *time-varying* functional connectivity, has useful predictive value for network operators. Even though functional connectivity is not effective (or causal) connectivity, a robust and comprehensive characterisation of correlation has long been established as an important step to *pinpoint the causes [of alarms] so that problems can be handled effectively* [87]. Functional connectivities can be thought of as spheres of influence such that in the presence of an incident, network operators (i) can rapidly discard events that might co-occur purely for spurious reasons, (ii) focus their effort on those nodes they know to be functionally related (including interpreting their content) and (iii) use this information to accelerate the process of fault localisation and root cause analysis. Importantly, inference of time-varying functional connectivity is crucial for dynamic computing environments where hardware, storage and network virtualisation enables the elastic provisioning of resources to a large and diverse set of services and applications. In such environments, service components (e.g. Docker containers or whole virtual machines) can be dynamically created, removed or migrated so that specific performance constraints are adhered to. We expect that functional connectivity inference will play a key role in fault analysis and prediction in such dynamic environments in the future.

Inferring functional connectivity is a hard problem. When using synthetic data for which ground truth was available, the F1-score only rarely exceeded 0.7 in the near-static case, 0.6 in the more dynamic case. However, this should not detract from the fact that the method was able to recover a substantial amount of the connectivity, including its changes over time, from an extremely limited amount of information. Indeed, it did so at least as well as state of the art methods in the near-static case, and usually better in the dynamic case. Importantly, unlike existing network inference methods (that typically do not handle sparse data well), it remains computationally tractable even with large networks (here, 10,000 nodes) over very long records (here, 10^7 observations). To be able to benchmark our method against state-of-the-art methods, we had to scale down to networks of size magnitudes smaller than our real-world application. The lack of scalability of these methods cannot be overstated.

Although our method produces weighted networks, where the weight denotes the strength of the interaction, in this chapter, we have been thresholding those weights throughout, both for prediction and evaluation purposes. Use of a threshold has a number of disadvantages, from losing important information about high-confidence edges (their distribution and organisation) to giving the same importance to high- and low-confidence edges. It also potentially confers the inference with sensitivity to the choice of the threshold. Whilst our experiments did not show evidence of such sensitivity (at least in the scenario that was tested), our results did show that the marginal gain in sensitivity due to using a small threshold came at the cost of a substantial drop in precision. An alternative is to use a continuous loss function based on the probabilities returned by our model. We are currently developing a method for automatically inferring the best threshold.

Our method returns a graph. It is reasonable to ask whether there is any reason for it other than visualisation. Whilst the visualisation aspect cannot be underestimated in the context of a network management system, we see the graph representation as an essential, albeit yet to be fully explored, component of the concept of functional connectivity in the sense of it being a starting point for understanding and analysing the system. In this chapter, we compared the properties of the (known) structural and (inferred) functional connectivities and were able to gain insights regarding the extent to which inferred functionality did capture an aspect of the events which was not included in the inference mechanism (namely, event types) as well as how structural connectivity underpinned connectivity between distinct functional connectivities. The operational implications of such insights remain to be seen. As we infer a time-varying connectivity, it will also be of interest to monitor how the characteristics of these inferred connectivities evolve over time. Indeed, it has been recently suggested that such analysis could help predict failures.

A key stumbling block in the development of this framework has been the absence of ground truth. Because functional connectivity only denotes statistical dependence (but not causal influence), there is no obvious way to provide an unequivocal assessment of how valid our inference is. In this work, we have used two approaches. One is to assess the extent to which the inferred model can predict future statistical dependences. Whilst this has operational value (e.g., filtering events based on knowledge that they are merely an expression of some latent statistical dependence to the activity of another node known to have emitted events), it is not an absolute measurement of quality since, for example, functional connectivity could be changing or nodes may not emit events during the period considered. Unsurprisingly, we have been consistently reporting low sensitivity values. The other approach is to use synthetic data. However, this has presented yet another challenge, namely, that of providing an accurate depiction of what happens in a live deployment. In this work, we have considered four types of network failure scenarios as proxy for functional connectivities. However, there are many other ways by which to define such connectivities. This highlights the need for controlled testbeds for experimentation. We are not aware of any and sadly there is little scope for experiments in commercial deployments, particularly when they involve critical services.

In conclusion, this is a first step toward developing the notion of dynamic functional connectivity inference in network management. To fulfil its full applicative potential, a more complete understanding of the various assumptions and parameters underpinning it must be obtained, which will be the subject of our future work.

Chapter 6

Conclusion

6.1 Summary of our work

The work presented in chapters 2, 3, 4 and 5 forms a continuous thread developing and validating a framework for inferring a functional connectivity from sparse time-series of events and of its application to computer networks.

The concept of functional connectivity is new to the field of computer network management. Hence, the challenges faced by anyone willing to infer a functional connectivity of a network of computers may be mysterious. Besides, the potential advantages brought by such a framework may not be immediately obvious. Finally, because most ISPs need to keep their network private for security reasons, there is very little work published about the events emitted by such networks and about the way they are exploited. Hence, as our funder provided us with a dataset from a major commercial company, within the second chapter of this thesis, we described in as much depth as possible the events emitted by this computer network as well as its topology. We observed that all the events emitted came with a textual description and we showed that it was possible to get a good idea of the function of almost every device using only this textual information. This was possible because we highlighted the fact that a large majority of devices emitted events of a single type and that event types were indicative of their function. Then, we showed that there was a significant number of pair of devices that emitted events in a temporally correlated manner and that among those, there was a significant number of pairs of devices that shared the same function. Hence these observations opened up the door for the inference of a functional topology from the time-series of events. However we also highlighted that the approach we used to measure a temporal correlation between time-series of events was not robust enough. Hence a functional network inference procedure able to deal with the challenges of such dataset was missing. The temporal analysis of the events indeed helped us pinpointing the obstacles to face. Firstly we observed that while most devices emitted very few events (≈2 per day), a small proportion of devices contributed to a significant proportion of the total number of events emitted. This was the main challenge: a large total number of events and a high sparsity of events from most nodes perspective. The second challenge laid in the duration of the recordings (millions of seconds) and in the size of the networks (ten thousands of devices). Finally, the last one laid in the potential change of the functional connectivity over time; devices are not expected to have the same unique function throughout their existence. We concluded this chapter by briefly highlighting the potential of a functional topology and indicating how it could help network management company make more sense of their events.

Within the third chapter we addressed all the challenges (but one that is addressed in the last chapter) that we highlighted within the second chapter. We delved into the mathematical aspect of the problem and created a pairwise coupling measures that (1) deals as well with sparse as with dense time-series, (2) that is able to detect non instantaneous interactions and (3) is easily scalable to large networks and long recordings. To do so, we started by considering a pair of independent Bernoulli processes of known rates and derived the exact analytical expressions of both the standard deviation and the expected value of the random variable measuring the number of pair of events (emitted by different processes) separated by a lag smaller than or equal to a given delay. We showed an excellent agreement between the analytical expressions and the empirical estimate of both the standard deviation and the expected value. We then proved and experimentally demonstrated that, in the limit of long recordings, for independent Bernoulli processes, this random variable is normally distributed. This enabled us to derive a Z-score and to interpret it as a coupling measure where a positive value provides evidence of statistical dependencies. We then considered limit cases and showed that this Z-score is robust to low sampling frequencies, i.e. to binned time-series. Besides, we showed it was able to measure coupling between Poisson processes. We also demonstrated robustness of the Z-score in the presence of auto-correlation (within limit). Further, we developed a delayed version of the common shock model and demonstrated an excellent ability of our Z-score to detect the presence of interactions even when recordings are short and/or sparse. Finally we compared our approach to two standard coupling measures, namely Pearson's cross-correlation and coherence, and showed that our Z-score is filling up a gap, i.e. that in specific contexts, it is better at detecting the presence of coupling than commonly used approaches.

However, estimating the functional topology from the application of the Z-score to every pair of nodes is not straightforward. Indeed, even in the absence of coupling, the Z-score may output values large by chance. We attempted to tackle this problem in the fourth chapter. Assuming that in the absence of interaction the statistics are distributed according to a normal distribution of unknown parameters, we developed a procedure that infers those parameters, the number of interacting pairs of nodes and the probability that each pair is interacting. This work aimed at providing a new tool for threshold decision making. Most statistically principled methods rely on false discovery rate procedure and limit the false positive rate to a predefined one. We argued that network inference procedure would greatly benefit from a thresholding procedure that would maximise a function of both false positive and false negative rates, e.g. a F1 score. Despite its obvious limitations (e.g. assumption of normality and necessity of small overlap between the distributions of interacting and non-interaction pair of nodes) we suggested that this approach is a first step toward such a goal. Nevertheless, we were not able to meaningfully apply this procedure to the dataset described in the second chapter because of the non-normality of the resulting statistic distribution. The Z-score is indeed only known to be normally distributed for Bernoulli processes. However, Bernoulli processes are too much of a simplification to describe the time-series of events. Hence, the obtained Z-score distribution is not normal and since this is a key assumption of the proposed method, we have not been able to apply it. This suggests that the normality assumption is too strong and that it should be relieved if we want the proposed method to be applied to real world datasets. This shows hence that the proposed method is a first step and that it will need to be adapted to non parametric distributions.

Within the fifth chapter, we developed a method that does not require to make such assumptions. The proposed approach divides the recording in windows, computes the statistics (developed in chapter 3) over every pair of nodes and over every window and infers a time-varying functional connectivity by maximising the model's predictive power from one window to the next. The output of the method is a weighted matrix where each edge weight denotes the strength of the statistical dependency between the nodes making up for the edge. This procedure overcomes all the previously mentioned challenges: it does not assume stationarity of the dynamics of the processes generating the time-series of events, it deals as well with sparse as with dense time-series of events, it is scalable to large networks and long recordings and it detects non instantaneous interactions. To the best of our knowledge there was no method that could deal with all these constraints.

To validate the accuracy of our prediction, we started by demonstrating its meaningfulness and underlying its potential. We highlighted in the second chapter that the type associated to each device was a relatively good proxy for each device's function. Hence in this chapter, we considered the relationship between edges weight and their nodes' type. We showed that the larger an edge weight is, the more likely the edge is to connect two nodes of the same type. Further we highlighted the fact that the higher the threshold we used to obtain a binary non-weighted network, the more likely we were to obtain, via a community detection algorithm, communities of nodes of the same type. Together these findings strongly suggest that our approach tends to group nodes of the same function. This highlights the functional nature of each detected community and hence pinpoints toward a way to identify subgroups of nodes providing the same service. Then we compared the structural topology to the one predicted by our approach and provided clear evidence of their complementarity. We showed that contrary to functional edges, structural edges tended to connect nodes of different types. This suggested that neither the predicted functional topology nor the guessed functional subgroups, could have been inferred with the sole knowledge of the structural topology.

Further, we compared, on synthetic and real world datasets, our approach to two state-of-the-art methods that came closest to ours. The first one provided something akin to a time-varying functional connectivity but was not specific to computer networks while the second one was computer network specific but only dealt with effective connectivity. We constructed the synthetic dataset so that it would mimic as closely as possible the dynamics and the topology of real world networks. We demonstrated the similarities between the datasets. Using synthetic datasets, we compared the ability of each method to recover the ground truth. Using real world datasets, we compared the predictive power of each approach, that is to say that we analysed their ability to predict the future from one window to the next. Results, on both type of datasets, highlighted the clear superiority of our approach. We also analysed the prediction made by our approach on the real-world datasets and demonstrated the robustness of our prediction over time. This enabled us to suggest a way to measure the presence of temporal changes of the functional connectivity over time. Finally, we highlighted the low computational cost of our approach by showing that our method was approximately 10^3 quicker than the two state-of-the-art methods already mentioned. We concluded this chapter by discussing the potential implications of this framework for computer network management and suggested ways by which outages identification might be sped up.

6.2 Implications

Within this thesis we demonstrated that it is possible to infer a meaningful functional connectivity from the time-series of events of computer network devices. This paradigm was imported from neuroscience where it enabled to get a deeper understanding of the dynamical mechanisms at stake within the the brain. One of the most important finding demonstrated that brain is not randomly structured. On the contrary, it is efficiently organised, combining local processing with global integration [117]. Other findings suggested that integration of information is facilitated by presence of large hubs regions. Others discovered that functionally connected regions tend to share same functions [160]. Note that this is also what we observed within computer networks. Functional brain analysis has also been applied to many conditions that are known to have deleterious effects such as ageing process [133], schizophrenia [48], Alzheimer's disease [149], Parkinson's disease [172]. Those analysis have been able to identify significative difference between functional topologies of conditioned and younger/healthy patients and thus help pinpointing the causes of such deleterious effects. For instance, age-related cognitive alterations are explained by a reduced functional connectivity in older subjects compared to young adults [133]. Hence those studies provide insights into the mechanisms at stake and can help pointing toward solutions to counteract negative effects.

Throughout this thesis we drew a parallel between brains and computer networks. Hence, we propose to extend the parallel to their functional connectivities. We argue that, as for brain networks, the functional connectivity is a tool that is designed to help network management companies understand their networks and outages that affect them. As we have seen, the functional topology differs a lot from the structural one. It is hence presenting a radically new perspective of the network and brings useful information that could not have been guessed otherwise. One of the application of this new paradigm lays in its ability to extract groups of devices that are involved in the provision of a particular service. We showed in the fifth chapter that this could indeed be done by applying community detection algorithm to the predicted functional topology. This is useful because it provides network operators with a higher level, and thus more comprehensive, representation of their network. Besides, in presence of an incident, it enables network operators to focus firstly their efforts on devices they know to be functionally related to the one indicating presence of an incident, thereby reducing the amount of devices to consider and thus the time needed to address incidents. Detection of such services is very challenging otherwise. Indeed the main other source of information is the textural description that comes with events. However, this text may not always be available, e.g. no meaningful types were reported with events in the second dataset of chapter 5, their informational content may remain too general (c.f. chapter 2), and/or they may involve templates manually entered, changing over time and subject to human error, thus rendering automatic classification difficult.

Further, the functional connectivity is useful because it complements knowledge gained from structural topologies. The gathering and maintenance of the structural topology is largely a manual process relying on probing tools, e.g. traceroute, ping... As a result, topologies are often a mixture of different layers and tend to be incomplete. As we have seen within the description of the dataset of the second chapter only one third of the devices emitting events were reported within the topology, while more than half of the devices known within the structural topology did not emit events. Besides, the structural topology was highly disconnected with many components made of a single device. Hence a potential application of the functional topology consists in demultiplexing layers within the structural topology. This is possible because each functional connected component may be seen a graph-representation of a given layer of interaction. Further, it can be used to functionally link different connected components of the structural network. We have indeed showed that functional and structural edges have very different behaviours. The first ones tend to link devices of the same type, while the second ones do the opposite. Many tools used to detect, analyse and help responding to faults rely on an accurate and complete description of the topology [94]. However, as we have highlighted previously, these topologies tend to be incomplete and inaccurate. The functional topology could hence be used to infer the gaps in understanding of the structure of the monitored network and as a consequence significantly improve the speed and accuracy of the fault management platform. Finally, we believe that the functional topology has useful predictive power. We indeed analysed our approach accuracy by measuring its predictive power and we listed some potential uses within the last chapter.

We believe nevertheless that this thesis impact should not remain limited to computer networks. We hope that our work will enable some advances to the robust inference of functional topology. While we observed a tendency of the field to move to theoretically motivated, parametric complex methods; we have been impressed at the same time by the good achievement of simple non-parametric pairwise coupling measure. This has been one conclusion resulting from the analysis of the DREAM competition, a competition involving dozens of procedures inferring functional networks: "The DREAM competition taught us some strong lessons. Amazingly, simpler methods performed in general better than more advanced, theoretically motivated approaches" [8]. Netoff et al. reached a similar conclusion in a paper analysing the ability to detect non linear coupling: "We have been as guilty as any of our colleagues in being fascinated by theory and methods of nonlinear dynamics. Hence we have continually been surprised by the robust capabilities of linear cross-correlation to detect weak coupling in non-linear systems, especially in the presence of noise" [108]. Within chapter 3, we followed this logic and developed a simple statistic that can help one measure the presence of coupling between two time-series of events. The approach extends the cross-correlation to its cumulated version. We showed that it is well suited to detect the presence of delayed coupling in the presence of sparse and short time-series of events and that within such context it is better at detecting the presence of coupling than commonly used methods. We hence hope that this method will help researchers detect presence of interaction when confronted to sparse time-series of events where delayed interactions is expected.

Finally, the remaining piece of work within this thesis has been focused on the determination of threshold for network inference. Whilst our method presents some obvious limitations, we consider it is as a first step toward the determination of a threshold based on both the false positive and false negative rates. We strongly believe that network analysis would greatly benefit from thresholding strategy that aim at maximising a function of those two rates and hence hope that this work will push toward this direction.

6.3 Future directions

The approach in its current form could be improved and there are many technical points that if addressed could greatly improve the predictive power of our approach.

Firstly, the approach described in the third chapter, relies on a comparison between processes of rates chosen to estimate the ones observed. However, when observing time-series of events, the number of events emitted is always known. Hence we are loosing information when not conditioning on the number of events really observed. This results in an increase of the variance of the statistics and thus in a decrease of the Z-score precision. Thus one future line of work consists in considering two independent random processes emitting a known number of uniformly distributed events. We have already derived the expected values and the standard deviation of the random variable counting the number of pair of events that are separated by a lag smaller than a given delay. However, the resulting random variable cannot be proven to be analytically normally distributed in the limit of long recordings and it is thus hard to get confidence levels. Nevertheless, empirically, it seems that this random variable is normally distributed. Hence, as future work we would like to analyse how robust is this empirical estimation and eventually apply it to real world dataset. This would strengthen our claim about the ability of the Z-score to measure efficiently the presence of delayed interactions. If the results are successful, it would also be interesting to extend our measure to its partial counterpart. Despite being relatively simple, the partial correlation has indeed proven to be a very powerful tool since, for instance, it enabled one team to win the first prize of the First Neural Connectomics Challenge in 2014 [150]. Another direction would be to extend the proposed functional statistics to its effective counterpart. Most commonly, the tools used to measure causality are coming from the field of information theory. However, as we have highlighted in the introduction, such information theoretic tools are currently badly suited to deal with sparse time-series of events. Hence, since it is relatively easy to adapt our statistic to only positive or negative delays, it would be interesting to analyse the performance of it. However this statistic would remain pairwise and as such would not be able to distinguish between direct and indirect interaction.

Another important issue with the Z-score developed in chapter 3 lays in the fact that a maximal lag needs to be a priori chosen to measure a deviation from statistical independence. When field knowledge is available, this may not be a problem. However in the general case, such maximal delay is hard to find. In the fifth chapter we worked around the problem by selecting the maximal lag that was chosen Kobayashi et al. [88]. There is however a source of information that has been overlooked that could solve this problem. If we consider a network inference problem, for which the Z-score is applied to every pair of nodes, the idea would be to choose the threshold that maximises the statistical distance between (1) the distribution of the Z-scores associated to a real edge and (2) the distribution of the Z-score associate with a non-existing edge. Although this remains to be examined thoroughly, it seems intuitive that the further apart the two distributions (edge and non-edge) are, the best the prediction is. This however assumes that we have been able to correctly characterise both distributions. This will hence only be possible if we manage to make the work started in chapter 4 more robust. Hence, a future line of work would consist in finding ways to overcome the two main limitations of the procedure proposed in chapter 4: (a) necessity of small overlap between the two distributions and (b) assumption of normality of the non-edge distribution. Preliminary work seems to show that the impact of the first limitation, i.e. (a), can be significantly decrease if we consider a different measure of success. Without delving too much in the details, instead of maximising the truncated normality of the truncated distribution, this would consist in maximising the normality of the reconstructed non-edge distribution. Furthermore, with the rise of computational power, it becomes easier to create surrogate data that closely resemble the observed dynamics and destroy statistical dependencies. Hence, a way to overcome the second issue, i.e. (b), could consist in carefully fitting a general analytical model to the distribution provided by the surrogate data to then perform a similar analysis. Combining those two aspects and thoroughly testing the framework remains challenging and provides an interesting future direction.

Finally, the approach developed in chapter 5 could also be improved. First, our analysis revealed that the proposed approach performs better for smaller networks than for larger ones. The most likely explanation lays in the fact that we used the same hyper-parameters for every pair of nodes. Those hyper-parameters are chosen by maximising the model's predictive power from one window to the next over every window and over every pair of nodes. Hence this suggests that the proposed method would benefit from several maximisations that would be done over smaller sets. It remains to be seen if different hyper-parameters could be selected for every pair of nodes. Furthermore those parameters are fixed over time, this implies that we expect the topological changes to remain more or less constant. Hence it would be interesting to investigate whether it is possible to make those hyper-parameters change over time and whether this would improve the precision of the method. Second, another problem lays in the choice of the maximal lag. Note that we provided a potential solution in the previous paragraph. Third, the last technical problem, we have been able to identify, lays within the choice of the windows size and within the way we cut the recording. Recent works proposed to use sliding windows [76] and it would hence be interesting to see wether such approach could be transposed to ours. Besides, because of windows overlap, the impact generated by uncertainty about the best window size is likely to be reduced.

Bibliography

[1] Theodore B Achacoso and William S Yamamoto. *AY's Neuroanatomy of C. elegans for Computation*. CRC Press, 1991.

[2] Genevera I Allen and Zhandong Liu. A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *2012 IEEE International Conference on Bioinformatics and Biomedicine*, pages 1–6. IEEE, 2012.

[3] Karen Appleby, G Goldszmidt, and Malgorzata Steinder. Yemanja-a layered event correlation engine for multi-domain server farms. In *Proc. of IFIP/IEEE IM*, 2001.

[4] D Arnett and TE Spraker. Cross-correlation analysis of the maintained discharge of rabbit retinal ganglion cells. *The Journal of Physiology*, 317(1):29–47, 1981.

[5] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9(Mar):485–516, 2008.

[6] Albert-Laszlo Barabasi. The new science of networks. 2002.

[7] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):60–69, 2003.

[8] Angela Baralla, Wieslawa I Mentzen, and Alberto De La Fuente. Inferring gene networks: dream or nightmare? part 1: Challenges 1 and 3. *Annals of the New York Academy of Sciences*, 1158(1):246–256, 2009.

[9] Janet Heine Barnett. Early writings on graph theory: Euler circuits and the königsberg bridge problem, 2005.

[10] Danielle S Bassett, Andreas Meyer-Lindenberg, Sophie Achard, Thomas Duke, and Edward Bullmore. Adaptive reconfiguration of fractal small-world human brain functional networks. *Proceedings of the National Academy of Sciences*, 103(51):19518–19523, 2006.

[11] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: series B (Methodological)*, 57(1):289–300, 1995.

[12] Ivan Beschastnikh, Yuriy Brun, Michael D Ernst, and Arvind Krishnamurthy. Inferring models of concurrent systems from logs of their behavior with csight. In *Proceedings of the 36th International Conference on Software Engineering*, pages 468–479. ACM, 2014.

[13] Stephan Bialonski, Martin Wendler, and Klaus Lehnertz. Unraveling spurious properties of interaction networks with tailored random networks. *PloS One*, 6(8):e22826, 2011.

[14] Patrick Billingsley. Probability and measure, theorem 27.5. *John Wiley&Sons, New York*, 1995.

[15] Katarzyna J Blinowska. Review of the methods of determination of directed connectivity from multichannel data. *Medical & Biological engineering & Computing*, 49(5):521–529, 2011.

[16] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[17] Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature Neuroscience*, 2(11):947, 1999.

[18] Radoslav Bortel and Pavel Sovka. Approximation of statistical distribution of magnitude squared coherence estimated with segment overlapping. *Signal Processing*, 87(5):1100–1117, 2007.

[19] Anastasios T Bouloutas, Seraphin Calo, and Allan Finkel. Alarm correlation and fault identification in communication networks. *IEEE Transactions on Communications*, 42(234):523–533, 1994.

[20] Steven L Bressler and Anil K Seth. Wiener–granger causality: a well established methodology. *Neuroimage*, 58(2):323–329, 2011.

[21] Carlos D Brody. Correlations without synchrony. *Neural Computation*, 11(7):1537–1551, 1999.

[22] Rowland Leonard Brooks. On colouring the nodes of a network. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 37, pages 194–197. Cambridge University Press, 1941.

[23] Emery N Brown, Robert E Kass, and Partha P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456, 2004.

[24] Ivan Brugere, Brian Gallagher, and Tanya Y Berger-Wolf. Network structure inference, a survey: Motivations, methods, and applications. *arXiv:1610.00782*, 2016.

[25] Ivan Brugere, Brian Gallagher, and Tanya Y Berger-Wolf. Network structure inference, a survey: Motivations, methods, and applications. *ACM Computing Surveys (CSUR)*, 51(2):24, 2018.

[26] Randy L Buckner, Jorge Sepulcre, Tanveer Talukdar, Fenna M Krienen, Hesheng Liu, Trey Hedden, Jessica R Andrews-Hanna, Reisa A Sperling, and Keith A Johnson. Cortical hubs revealed by intrinsic functional connectivity: mapping, assessment of stability, and relation to alzheimer's disease. *Journal of Neuroscience*, 29(6):1860–1873, 2009.

[27] György Buzsáki. Large-scale recording of neuronal ensembles. *Nature Neuroscience*, 7(5):446, 2004.

[28] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468, 2000.

[29] JT Campbell. The poisson correlation function. *Proceedings of the Edinburgh Mathematical Society*, 4(1):18–26, 1934.

[30] Victor Carluccio, Nidhal Bouaynaya, Gregory Ditzler, and Hassan M Fathallah-Shaykh. The akron-kalman filter for tracking time-varying networks. In *Proc. of IEEE BHI*, 2017.

[31] Zhe Chen. An overview of Bayesian methods for neural spike train analysis. *Computational Intelligence and Neuroscience*, 2013:1, 2013.

[32] Zhe Chen, David F Putrino, Soumya Ghosh, Riccardo Barbieri, and Emery N Brown. Statistical inference for assessing functional connectivity of neuronal ensembles with

sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):121–135, 2010.

[33] Zhe Chen, David F Putrino, Soumya Ghosh, Riccardo Barbieri, and Emery N Brown. Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):121–135, 2011.

[34] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[35] Luciano da Fontoura Costa, Osvaldo N Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira, Matheus Palhares Viana, and Luis Enrique Correa Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, 2011.

[36] David R Cox and Peter Adrian Walter Lewis. Multivariate point processes. In *Proc. 6th Berkeley Symp. Math. Statist. Prob*, volume 3, pages 401–448, 1972.

[37] Ildefons Magrans de Abril, Junichiro Yoshimoto, and Kenji Doya. Connectivity inference from neural recording data: Challenges, mathematical bases and research directions. *Neural Networks*, 102:120–137, 2018.

[38] Frank Dondelinger, Sophie Lèbre, and Dirk Husmeier. Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, 90(2):191–230, 2013.

[39] JJ Eggermont. Neural interaction in cat primary auditory cortex. dependence on recording depth, electrode separation, and age. *Journal of Neurophysiology*, 68(4):1216–1228, 1992.

[40] Jos J Eggermont. The correlative brain. In *The correlative brain*, pages 267–281. Springer, 1990.

[41] Victor M Eguiluz, Dante R Chialvo, Guillermo A Cecchi, Marwan Baliki, and A Vania Apkarian. Scale-free brain functional networks. *Physical Review Letters*, 94(1):018102, 2005.

[42] Farnaz Zamani Esfahlani and Hiroki Sayama. A percolation-based thresholding method with applications in functional connectivity analysis. In *International Workshop on Complex Networks*, pages 221–231. Springer, 2018.

[43] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

[44] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[45] Mark Fiecas, Hernando Ombao, Crystal Linkletter, Wesley Thompson, and Jerome Sanes. Functional connectivity: Shrinkage estimation and randomization test. *NeuroImage*, 49(4):3005–3014, 2010.

[46] Ronald A Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

[47] Ronald A Fisher. On the probable error of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

[48] Jennifer Fitzsimmons, Marek Kubicki, and Martha E Shenton. Review of functional and anatomical brain connectivity findings in schizophrenia. *Current Opinion in Psychiatry*, 26(2):172–187, 2013.

[49] Karl Friston and Will Penny. Post hoc Bayesian model selection. *Neuroimage*, 56(4):2089–2099, 2011.

[50] Karl J Friston. Functional and effective connectivity: a review. *Brain Connectivity*, 1(1):13–36, 2011.

[51] Clément Gallet and Claude Julien. The significance threshold for coherence when using the welch's periodogram method: effect of overlapping segments. *Biomedical Signal Processing and Control*, 6(4):405–409, 2011.

[52] Kathleen A Garrison, Dustin Scheinost, Emily S Finn, Xilin Shen, and R Todd Constable. The (in) stability of functional brain network measures across thresholds. *Neuroimage*, 118:651–661, 2015.

[53] Aaron Gember-Jacobson, Wenfei Wu, Xiujun Li, Aditya Akella, and Ratul Mahajan. Management plane analytics. In *Proceedings of the 2015 Internet Measurement Conference*, pages 395–408. ACM, 2015.

[54] Christian Genest, Mhamed Mesfioui, and Juliana Schulz. A new bivariate poisson common shock model covering all possible degrees of dependence. *Statistics & Probability Letters*, 140:202–209, 2018.

[55] GL Gerstein and WA Clark. Simultaneous studies of firing patterns in several neurons. *Science*, 143(3612):1325–1327, 1964.

[56] Cedric E Ginestet and Andrew Simmons. Statistical parametric network analysis of functional connectivity dynamics during a working memory task. *Neuroimage*, 55(2):688–704, 2011.

[57] Roy J Glauber. Time-dependent statistics of the ising model. *Journal of Mathematical Physics*, 4(2):294–307, 1963.

[58] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. Evolve or die: High-availability design principles drawn from googles network infrastructure. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 58–72. ACM, 2016.

[59] Aslak Grinsted, John C Moore, and Svetlana Jevrejeva. Application of the cross wavelet transform and wavelet coherence to geophysical time series. *Nonlinear Processes in Geophysics*, 11(5/6):561–566, 2004.

[60] Sonja Grün, Markus Diesmann, and Ad Aertsen. Unitary events in multiple single-neuron spiking activity: I. detection and significance. *Neural Computation*, 14(1):43–80, 2002.

[61] Sonja Grün, Markus Diesmann, and Ad Aertsen. Unitary events in multiple single-neuron spiking activity: Ii. nonstationary data. *Neural Computation*, 14(1):81–119, 2002.

[62] Sonja Grün, Markus Diesmann, Franck Grammont, Alexa Riehle, and Ad Aertsen. Detecting unitary events without discretization of time. *Journal of Neuroscience Methods*, 94(1):67–79, 1999.

[63] Sonja Grün. Data-driven significance estimation for precise spike correlation. *Journal of Neurophysiology*, 101(3):1126–1140, 2009.

[64] Fan Guo, Steve Hanneke, Wenjie Fu, and Eric P Xing. Recovering temporally rewiring networks: A model-based approach. In *Proc. of ICML*, 2007.

[65] David Hallac, Youngsuk Park, Stephen Boyd, and Jure Leskovec. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 205–213. ACM, 2017.

[66] David M Halliday. Nonparametric directionality measures for time series and point process data. *Journal of Integrative Neuroscience*, 14(02):253–277, 2015.

[67] David M Halliday, John-Stuart Brittain, Carl W Stevenson, and Rob Mason. Adaptive spectral tracking for coherence estimation: the z-tracker. *Journal of Neural Engineering*, 15(2):026004, 2018.

[68] David M Halliday and Jay R Rosenberg. Time and frequency domain analysis of spike train and time series data. In *Modern techniques in neuroscience research*, pages 503–543. Springer, 1999.

[69] DM Halliday, JR Rosenberg, AM Amjad, P Breeze, BA Conway, and SF Farmer. A framework for the analysis of mixed time series/point process data-theory and application to the study of physiological tremor, single motor unit discharges and electromyograms. *Progress in Biophysics and Molecular Biology*, 64(2):237, 1995.

[70] Robert Harper and Philip Tee. The application of neural networks to predicting the root cause of service failures. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 953–958. IEEE, 2017.

[71] Robert Harper and Philip Tee. A method for temporal event correlation. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 13–18. IEEE, 2019.

[72] Masum Hasan, Binay Sugla, and Ramesh Viswanathan. A conceptual framework for network management event correlation and filtering systems. In *Proc. of IFIP/IEEE IM*, 1999.

[73] Satoru Hayasaka and Paul J Laurienti. Comparison of characteristics between region- and voxel-based network analyses in resting-state fmri data. *Neuroimage*, 50(2):499–508, 2010.

[74] Shilin He, Jieming Zhu, Pinjia He, and Michael R Lyu. Experience report: system log analysis for anomaly detection. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pages 207–218. IEEE, 2016.

[75] Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren, and Reinhard Guthke. Gene regulatory network inference: data integration in dynamic models?a review. *Biosystems*, 96(1):86–103, 2009.

[76] Rikkert Hindriks, Mohit H Adhikari, Yusuke Murayama, Marco Ganzetti, Dante Mantini, Nikos K Logothetis, and Gustavo Deco. Can sliding-window correlations reveal dynamic functional connectivity in resting-state fmri? *Neuroimage*, 127:242–256, 2016.

[77] Conor Houghton. Calculating the mutual information between two spike trains. *Neural Computation*, 31(2):330–343, 2019.

[78] Mahdi Jalili. Functional brain networks: does the choice of dependency estimator and binarization method matter? *Scientific Reports*, 6:29780, 2016.

[79] Mahdi Jalili and Maria G Knyazeva. Eeg-based functional networks in schizophrenia. *Computers in Biology and Medicine*, 41(12):1178–1186, 2011.

[80] MR Jarvis and PP Mitra. Sampling properties of the spectrum and coherency of sequences of action potentials. *Neural Computation*, 13(4):717–749, 2001.

[81] DH Johnson and NY Kiang. Analysis of discharges recorded simultaneously from pairs of auditory nerve fibers. *Biophysical Journal*, 16(7):719–734, 1976.

[82] Sam Kash Kachigan. *Statistical analysis: An interdisciplinary introduction to univariate & multivariate methods.* Radius Press, 1986.

[83] Stefan Kätker and Martin Paterok. Fault isolation and event correlation for integrated fault management. In *Proc. of IFIP/IEEE IM*, 1997.

[84] Irene Katzela and Mischa Schwartz. Schemes for fault identification in communication networks. *IEEE/ACM Transactions on Networking*, 3(6):753–764, 1995.

[85] Jehandad Khan, Nidhal Bouaynaya, and Hassan M Fathallah-Shaykh. Tracking of time-varying genomic regulatory networks with a lasso-kalman smoother. *EURASIP Journal on Bioinformatics and Systems Biology*, 2014(1):3, 2014.

[86] Joshua Kinnison, Srikanth Padmala, Jong-Moon Choi, and Luiz Pessoa. Network analysis reveals increased integration during emotional and motivational processing. *Journal of Neuroscience*, 32(24):8361–8372, 2012.

[87] Shmuel Kliger, Shaula Yemini, Yechiam Yemini, David Ohsie, and Salvatore Stolfo. A coding approach to event correlation. In *Proc. of IFIP/IEEE IM*, 1995.

[88] Satoru Kobayashi, Kensuke Fukuda, and Hiroshi Esaki. Mining causes of network events in log data with causal inference. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 45–53. IEEE, 2017.

[89] Mark A Kramer, Uri T Eden, Sydney S Cash, and Eric D Kolaczyk. Network inference with confidence from multivariate time series. *Physical Review E*, 79(6):061916, 2009.

[90] Mark A Kramer, Eric D Kolaczyk, and Heidi E Kirsch. Emergent network topology at seizure onset in humans. *Epilepsy Research*, 79(2-3):173–186, 2008.

[91] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600. AcM, 2010.

[92] Jean-Philippe Lachaux, Antoine Lutz, David Rudrauf, Diego Cosmelli, Michel Le Van Quyen, Jacques Martinerie, and Francisco Varela. Estimating the time-course of coherence between single-trial brain signals: an introduction to wavelet coherence. *Neurophysiologie Clinique/Clinical Neurophysiology*, 32(3):157–174, 2002.

[93] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701, 2001.

[94] Ma łgorzata Steinder and Adarshpal S Sethi. A survey of fault localization techniques in computer networks. *Science of computer programming*, 53(2):165–194, 2004.

[95] Qingyun Liu, Shiliang Tang, Xinyi Zhang, Xiaohan Zhao, Ben Y Zhao, and Haitao Zheng. Network growth and link prediction through an empirical lens. In *Proc. of ACM IMC*, 2016.

[96] Sebastien Louis, Christian Borgelt, and Sonja Grün. Generation and selection of surrogate methods for correlation analysis. In *Analysis of Parallel Spike Trains*, pages 359–382. Springer, 2010.

[97] Sebastien GR Louis, Markus Diesmann, et al. Surrogate spike train generation through dithering in operational time. *Frontiers in Computational Neuroscience*, 4:127, 2010.

[98] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[99] Ajay Anil Mahimkar, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Qi Zhao. Towards automated performance diagnosis in a large IPTV network. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 231–242, 2009.

[100] Ajay Anil Mahimkar, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Qi Zhao. Towards automated performance diagnosis in a large iptv network. In *Proc. of SIGCOMM*, 2009.

[101] Ed McKenzie. Some simple models for discrete variate time series. *JAWRA Journal of the American Water Resources Association*, 21(4):645–650, 1985.

[102] Antoine Messager, Nicos Georgiou, and Luc Berthouze. A new method for the robust characterisation of pairwise statistical dependency between point processes. *arXiv preprint arXiv:1904.04813*, 2019.

[103] Antoine Messager, George Parisis, Robert Harper, Philip Tee, István Z Kiss, and Luc Berthouze. Network events in a large commercial network: What can we learn? In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2018.

[104] Sifis Micheloyannis, Ellie Pachou, Cornelis Jan Stam, Michael Breakspear, Panagiotis Bitsios, Michael Vourkas, Sophia Erimaki, and Michael Zervakis. Small-world networks and disturbed functional connectivity in schizophrenia. *Schizophrenia Research*, 87(1-3):60–66, 2006.

[105] Melanie Mitchell. *Complexity: A guided tour.* Oxford University Press, 2009.

[106] Ricardo Pio Monti, Peter Hellyer, David Sharp, Robert Leech, Christoforos Anagnostopoulos, and Giovanni Montana. Estimating time-varying brain connectivity networks from functional mri time series. *NeuroImage*, 103:427–443, 2014.

[107] M Müller, G Baier, C Rummel, and K Schindler. Estimating the strength of genuine and random correlations in non-stationary multivariate time series. *EPL (Europhysics Letters)*, 84(1):10009, 2008.

[108] Theoden I Netoff, Thomas L Carroll, Louis M Pecora, and Steven J Schiff. Detecting coupling in the presence of noise and nonlinearity. In *Handbook of Time Series*

Analysis: Recent Theoretical Developments and Applications, pages 265–282. Wiley-VCH Verlag GmbH & Co. KGaA, 2006.

[109] Theoden I Netoff, Louis M Pecora, and Steven J Schiff. Analytical coupling detection in the presence of noise and nonlinearity. *Physical Review E*, 69(1):017201, 2004.

[110] Mark Newman. *Networks*. Oxford university press, 2018.

[111] Mark EJ Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, 2002.

[112] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[113] Murat Okatan, Matthew A Wilson, and Emery N Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation*, 17(9):1927–1961, 2005.

[114] Adam J. Oliner, Ashutosh V. Kulkarni, and Alex Aiken. Using correlated surprise to infer shared influence. *Proceedings of the International Conference on Dependable Systems and Networks*, pages 191–200, 2010.

[115] Alina Oprea, Zhou Li, Ting-Fang Yen, Sang H Chin, and Sumayah Alrwais. Detection of early-stage enterprise infection by mining large-scale log data. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 45–56. IEEE, 2015.

[116] G Palm, AMHJ Aertsen, and GL Gerstein. On the significance of correlations among neuronal spike trains. *Biological Cybernetics*, 59(1):1–11, 1988.

[117] Hae-Jeong Park and Karl Friston. Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411, 2013.

[118] Roberto D Pascual-Marqui. Instantaneous and lagged measurements of linear and nonlinear dependence between groups of multivariate time series: frequency decomposition. *arXiv preprint arXiv:0711.1455*, 2007.

[119] Ahmed Patel, Gabriel McDermott, and Conn Mulvihill. Integrating network management and artificial intelligence. In *Proc. of IFIP/IEEE IM*, 1989.

[120] Gordon Pipa, Sonja Grün, and Carl Van Vreeswijk. Impact of spike train autostructure on probability distribution of joint spike events. *Neural Computation*, 25(5):1123–1163, 2013.

[121] Rahul Potharaju and Navendu Jain. Demystifying the dark side of the middle: a field study of middlebox failures in datacenters. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, pages 9–22. ACM, 2013.

[122] Pietro Quaglio, Vahid Rostami, Emiliano Torre, and Sonja Grün. Methods for identification of spike patterns in massively parallel spike trains. *Biological Cybernetics*, 112(1-2):57–80, 2018.

[123] Joshua W Robinson and Alexander J Hartemink. Learning non-stationary dynamic Bayesian networks. *Journal of Machine Learning Research*, 11(Dec):3647–3680, 2010.

[124] Manuel Gomez Rodriguez, Jure Leskovec, David Balduzzi, and Bernhard Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2(1):26–65, 2014.

[125] Yasser Roudi, Benjamin Dunn, and John Hertz. Multi-neuronal activity and functional connectivity in cell assemblies. *Current Opinion in Neurobiology*, 32:38–44, 2015.

[126] John P Rouillard. Real-time log file analysis using the simple event correlator (sec). In *LISA*, volume 4, pages 133–150, 2004.

[127] Arup Roy, Peter N. Steinmetz, and Ernst Niebur. Rate limitations of unitary event analysis. *Neural Computation*, 12(9):2063–2082, 2000.

[128] Sudip Roy, Arnd Christian König, Igor Dvorkin, and Manish Kumar. Perfaugur: Robust diagnostics for performance anomalies in cloud services. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1167–1178. IEEE, 2015.

[129] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

[130] Mikail Rubinov and Olaf Sporns. Weight-conserving characterization of complex functional brain networks. *Neuroimage*, 56(4):2068–2079, 2011.

[131] Christian Rummel, Markus Müller, Gerold Baier, Frédérique Amor, and Kaspar Schindler. Analyzing spatio-temporal patterns of genuine cross-correlations. *Journal of Neuroscience Methods*, 191(1):94–100, 2010.

[132] R Saab, MJ McKeown, LJ Myers, and R Abu-Gharbieh. A wavelet based approach for the detection of coupling in eeg signals. In *Conference Proceedings. 2nd International IEEE EMBS Conference on Neural Engineering, 2005.*, pages 616–620. IEEE, 2005.

[133] Roser Sala-Llonch, David Bartrés-Faz, and Carme Junqué. Reorganization of brain networks in aging: a review of functional connectivity studies. *Frontiers in Psychology*, 6:663, 2015.

[134] Elad Schneidman, Michael J Berry II, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007, 2006.

[135] Thomas Schreiber. Measuring information transfer. *Physical Review Letters*, 85(2):461, 2000.

[136] TA Sears and D Stagg. Short-term synchronization of intercostal motoneurone activity. *The Journal of Physiology*, 263(3):357–381, 1976.

[137] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[138] Xuesi Shao and Peixi Chen. Normalized auto-and cross-covariance functions for neuronal spike train analysis. *International Journal of Neuroscience*, 34(1-2):85–95, 1987.

[139] Jonathon Shlens. Notes on generalized linear models of neurons. *arXiv preprint arXiv:1404.1999*, 2014.

[140] Sean L Simpson, F DuBois Bowman, and Paul J Laurienti. Analyzing complex functional brain networks: fusing statistics and network science to understand the brain. *Statistics Surveys*, 7:1, 2013.

[141] Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

[142] Dong Song, Haonan Wang, Catherine Y Tu, Vasilis Z Marmarelis, Robert E Hampson, Sam A Deadwyler, and Theodore W Berger. Identification of sparse neural functional connectivity using penalized likelihood estimation and basis functions. *Journal of Computational Neuroscience*, 35(3):335–357, 2013.

[143] Le Song, Mladen Kolar, and Eric P Xing. Time-varying dynamic Bayesian networks. In *Advances in Neural Information Processing Systems*, pages 1732–1740, 2009.

[144] Olaf Sporns. Brain connectivity. *Scholarpedia*, 2(10):4695, 2007.

[145] Olaf Sporns. *Networks of the Brain*. MIT press, 2010.

[146] Ian H Stevenson, James M Rebesco, Nicholas G Hatsopoulos, Zach Haga, Lee E Miller, and Konrad P Kording. Bayesian inference of functional connectivity and network structure from spikes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(3):203–213, 2008.

[147] Ian H Stevenson, James M Rebesco, Nicholas G Hatsopoulos, Zach Haga, Lee E Miller, and Konrad P Kording. Bayesian inference of functional connectivity and network structure from spikes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(3):203–213, 2009.

[148] Steven P Strong, Roland Koberle, Rob R De Ruyter Van Steveninck, and William Bialek. Entropy and information in neural spike trains. *Physical Review Letters*, 80(1):197, 1998.

[149] Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael D Greicius. Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS Computational Biology*, 4(6):e1000100, 2008.

[150] Antonio Sutera, Arnaud Joly, Vincent François-Lavet, Aaron Qiu, Gilles Louppe, Damien Ernst, and Pierre Geurts. Simple connectome inference from partial correlation statistics in calcium imaging. In *Neural Connectomics Workshop*, pages 23–35, 2015.

[151] Philip Tee, George Parisis, and Ian Wakeman. Vertex entropy as a critical node measure in network monitoring. *IEEE Transactions on Network and Service Management*, 14(3):646–660, 2017.

[152] Philip Tee, Ian Wakeman, George Parisis, Jonathan Dawes, and István Z Kiss. Constraints and entropy in a model of network evolution. *The European Physical Journal B*, 90(11):226, 2017.

[153] Prejaas Tewarie, Edwin van Dellen, Arjan Hillebrand, and Cornelis J Stam. The minimum spanning tree: an unbiased method for brain network analysis. *Neuroimage*, 104:177–188, 2015.

[154] Cibu Thomas, Q Ye Frank, M Okan Irfanoglu, Pooja Modi, Kadharbatcha S Saleem, David A Leopold, and Carlo Pierpaoli. Anatomical accuracy of brain connections derived from diffusion mri tractography is inherently limited. *Proceedings of the National Academy of Sciences*, 111(46), 2014.

[155] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[156] Anastasios A Tsonis and Paul J Roebber. The architecture of the climate network. *Physica A: Statistical Mechanics and its Applications*, 333:497–504, 2004.

[157] John W Tukey. Exploratory data analysis. pages 39–49, 1977.

[158] Liubov Tupikina, Nora Molkenthin, Cristóbal López, Emilio Hernández-García, Norbert Marwan, and Jürgen Kurths. Correlation networks from flows. the case of forced and time-dependent advection-diffusion dynamics. *PloS One*, 11(4):e0153703, 2016.

[159] Martijn P van den Heuvel, Siemon C de Lange, Andrew Zalesky, Caio Seguin, BT Thomas Yeo, and Ruben Schmidt. Proportional thresholding in resting-state fmri functional connectivity networks and consequences for patient-control connectome studies: Issues and recommendations. *Neuroimage*, 152:437–449, 2017.

[160] Martijn P Van Den Heuvel and Hilleke E Hulshoff Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European Neuropsychopharmacology*, 20(8):519–534, 2010.

[161] Martijn P van den Heuvel and Olaf Sporns. Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–696, 2013.

[162] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[163] Jinhui Wang, Liang Wang, Yufeng Zang, Hong Yang, Hehan Tang, Qiyong Gong, Zhang Chen, Chaozhe Zhu, and Yong He. Parcellation-dependent small-world brain functional networks: a resting-state fmri study. *Human Brain Mapping*, 30(5):1511–1523, 2009.

[164] Zhaowen Wang, Ercan E Kuruoglu, Xiaokang Yang, Yi Xu, and Thomas S Huang. Time varying dynamic Bayesian network for nonstationary events modeling and online inference. *IEEE Transactions on Signal Processing*, 59(4):1553–1568, 2011.

[165] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[166] Geoffrey B West, James H Brown, and Brian J Enquist. The fourth dimension of life: fractal geometry and allometric scaling of organisms. *science*, 284(5420):1677–1679, 1999.

[167] Wikipedia. `https://en.wikipedia.org/wiki/File:Konigsberg_bridges.png`.

[168] Wikipedia. `https://en.wikipedia.org/wiki/File:7_bridges.svg`.

[169] Wikipedia. `https://en.wikipedia.org/wiki/File:K%C3%B6nigsberg_graph.svg`.

[170] Wikipedia. `https://commons.wikimedia.org/wiki/File:Chemical_synapse_schema_cropped.jpg`.

[171] Ernst C Wit and Antonino Abbruzzo. Inferring slowly-changing dynamic gene-regulatory networks. *BMC Bioinformatics*, 16(6):S5, 2015.

[172] Tao Wu, Xiangyu Long, Liang Wang, Mark Hallett, Yufeng Zang, Kuncheng Li, and Piu Chan. Functional connectivity of cortical motor areas in the resting state in parkinson's disease. *Human Brain Mapping*, 32(9):1443–1457, 2011.

[173] Kenji Yamanishi and Yuko Maruyama. Dynamic syslog mining for network failure monitoring. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 499–508. ACM, 2005.

[174] Kazuko Yamasaki, Avi Gozolchiani, and Shlomo Havlin. Climate networks around the globe are significantly affected by el nino. *Physical Review Letters*, 100(22):228501, 2008.

[175] Fang-Chin Yeh, Aonan Tang, Jon Hobbs, Pawel Hottowy, Wladyslaw Dabrowski, Alexander Sher, Alan Litke, and John Beggs. Maximum entropy approaches to living neural networks. *Entropy*, 12(1):89–106, 2010.

[176] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 199–208. ACM, 2013.

[177] Xiao Yu, Pallavi Joshi, Jianwu Xu, Guoliang Jin, Hui Zhang, and Guofei Jiang. Cloudseer: Workflow monitoring of cloud infrastructures via interleaved logs. In *ACM SIGPLAN Notices*, volume 51, pages 489–502. ACM, 2016.

[178] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[179] Andrew Zalesky, Alex Fornito, and Ed Bullmore. On the use of correlation as a measure of network connectivity. *Neuroimage*, 60(4):2096–2106, 2012.

[180] Amin Zandvakili and Adam Kohn. Coordinated neuronal activity enhances cortico-cortical communication. *Neuron*, 87(4):827–839, 2015.

[181] Ziming Zheng, Li Yu, Zhiling Lan, and Terry Jones. 3dimensionall root cause diagnosis via co-analysis. In *Proc. of ICAC*, 2012.

[182] Shuheng Zhou, John Lafferty, and Larry Wasserman. Time varying undirected graphs. *Machine Learning*, 80(2):295–319, 2010.