Temporal Scale of Dynamic Networks

BY

RAJMONDA SULO CÁCERES

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2012

Chicago, Illinois

Defense Committee:

Tanya Berger-Wolf, Chair and Advisor, *Department of Computer Science*
Gyorgy Turan, *Department of Mathematics, Statistics and Computer Science*
Jie Yang, *Department of Mathematics, Statistics and Computer Science*
Robert Grossman, *University of Chicago*
Michael Pelsmayer, *Illinois Institute of Technology*
Hemanshu Kaul, *Illinois Institute of Technology*

Dedicated to my father Zalo Sulo.

Babi, faleminderit për gjithshka.

Shpresoj të jesh krenar për mua.

ACKNOWLEDGMENTS

I would like to thank many people who helped make this thesis possible. First and foremost, I would like to thank my advisor, Tanya Berger-Wolf. It has a been a great privilege to call her my advisor and my mentor. She has generously shared her knowledge and experience, always cared to give the hard advice as well as her ongoing encouragement, but most of all, reminded me constantly of the excitement that comes with doing research and the opportunity to make an impact. I would like to thank the members of my committee for their time, support and useful suggestions: Gyorgy Turan, Jie Yang, Robert Grossman, Michael Pelsmayer and Hemanshu Kaul. I would like to thank my friends with whom I have shared my PhD journey: Mechie, Despina, Anushka, Nitin, Paul, Saad, Chayant, Habiba, Mayank, Arun, Marco, Islam, Dimitri, Maria, Jenni, Khairi and Victor. I am very grateful to George and Janet for becoming like a second family to me and offering a much needed support system. To Dan, Mosheh and Vijay, thank you for all the invaluable advice and interesting discussions. I would also like to thank the following people that at one point or another have helped me navigate the intricacies of the PhD process: Kari Dueball, Maureen Madden, Silvia Becerra, Siim Soot, Ashish Sen, Paul Metaxatos, Veronica Arreola, Sarah Shirk, Cindy Rogowski, Shirley Connelly, David Turkington, Paul Coe, Jeannette Ollis. I am very indebted and forever grateful to my family for supporting my dreams every step of the way, my parents Zalo and Merjeme, my brothers Miri and Taku. To my mother-in-law Lidia and father-in-law Rigoberto, thank you all the love and support. A special thank

you goes to the kids in my life, that so effortlessly brought smiles and brightened my days: Mary, Elina, Eva and Margaret. Finally, I feel very lucky and humbled to have shared this journey with my husband and my best friend, Giovanni. His support, understanding, and unconditional love has been a constant source of strength and motivation, and also a big reason I can write this letter today.

TABLE OF CONTENTS

CHAPTER			PAGE
1	**INTRODUCTION**		1
	1.1	Overview	1
	1.1.1	The Dynamic Network Abstraction	3
	1.1.2	Empirical Motivation	5
	1.1.3	Data Collection	5
	1.1.4	Oversampling of Temporal Streams	8
	1.2	Main Contributions and Findings	10
	1.3	Thesis Outline	11
	1.4	List of Publications and Manuscripts	12
2	**RELATED WORK**		13
	2.1	Signal Processing & Information Theory	13
	2.2	Time Series Smoothing	14
	2.3	Temporal Scale of Dynamic Networks	14
3	**PRELIMINARIES AND PROBLEM FORMULATION**		17
	3.1	Definitions and Notation	17
	3.2	Problem Formulation	22
4	**EXAMPLES AND SPECIAL CASES**		23
	4.1	Constant Streams	23
	4.2	Random Unstructured Streams	24
	4.2.1	$DynUR$ Streams	24
	4.2.1.1	Stationarity of functions on $DynUR$ Streams	26
	4.2.1.2	Temporal Order Invariance of $DynUR$ Streams	28
	4.3	Structured Temporal Streams	32
	4.3.1	Periodic Streams	32
	4.3.2	Stationary Streams	37
	4.3.3	Theseus Ship Streams	38
	4.3.4	Oversampled Streams	40
	4.3.5	Noisy Streams	43
	4.3.6	Oversampled, Noisy Stationary Streams	44
	4.3.7	Real-World Streams	47
	4.4	Experimental Results	47
	4.5	Analytical Framework for the TSI problem	50
5	**AXIOMATIC FRAMEWORK**		52

TABLE OF CONTENTS (Continued)

CHAPTER		PAGE
5.1	Axiomatic Framework: desired properties of the quality function Q	53
5.2	Discussion	59
6	**INHERENT TIMESCALE**	61
6.1	Information Theoretic Approach	61
6.1.1	TWIN Heuristic	62
6.1.2	Experimental Setup	66
6.1.3	Experimental Results	67
6.1.3.1	Real-world Dynamic Networks	67
6.1.3.2	Synthetic Dynamic Networks	68
6.1.3.3	Comparison with Graphscope and FFT Method.	70
6.1.4	TWIN in the Context of the Axiomatic Framework	73
6.1.4.1	Perturbation Analysis	73
6.1.4.2	Oversampling Analysis	76
6.1.5	Summary	77
6.2	Persistence-based Approach	81
6.2.1	DAPPER Heuristic	82
6.2.1.1	Measuring Local Persistence	82
6.2.2	Outline of the DAPPER Heuristic	88
6.2.3	Experimental Setup	89
6.2.4	Experimental Results	90
6.2.5	Discussion	93
6.2.5.1	Challenges for Larger Values of ω	93
6.2.5.2	Desired Properties of a Local Quality Measure	96
6.2.5.3	Comparison of the DAPPER Heuristic with TWIN and Graphscope Heuristics	98
7	**FUTURE WORK AND CONCLUSIONS**	100
7.1	Directions for Future Work	100
7.1.1	Analysis of Special Cases of Interaction Streams	100
7.1.2	Extensions to Perturbation Analysis Framework	101
7.1.3	Objective-based Formulations of TSI Problem	102
7.1.3.1	Algorithmic-specific Formulation of TSI Problem	103
7.1.3.2	TSI Problem for Dynamic Community Identification	103
7.1.3.3	TSI Problem for Dynamic Link Prediction	104
7.2	Conclusions	105
	CITED LITERATURE	107
	VITA	113

LIST OF FIGURES

FIGURE		PAGE
1	Representation of a social network aggregated at different windows of aggregation (a), and network measures of the Reality Mining dataset as functions of the window of aggregation (b).	6
2	Illustration of two different representations of time in interaction streams.	7
3	Aggregation of a temporal interaction stream with window of aggregation $\omega = 3$.	19
4	Aggregation of a static stream at different levels of aggregation.	24
5	Time Series of a function f computed over a dynamic network. .	26
6	Illustration of two cases of the $DynMix$ graph with three alternating probability distributions (M=3). Subfigure (a) illustrates Case 1 of $DynMix$ with a cycle of constant probabilities for each edge, and Subfigure (b) illustrates Case 2 of $DynMix$ with a cycle of Erdös-Rényi graphs.	33
7	Aggregation of Theseus Ship stream at different scales. The original Theseus Ship stream has no interactions as the base set and one interaction changes every time step.	39
8	Oversampled periodic stream.	41
9	Variance of the network density measure as a function of the window of aggregation for the $DynUR$, $DynMix$, Reality Mining and Haggle datasets.	49
10	Analytical framework for analyzing the temporal scale of interaction streams.	50
11	Illustration of computation of variance and compression ratio for a given level of aggregation (ω), and a given graph-theoretic measure d (diameter).	63
12	Trade-off plot of noise and compression measures.	64
13	TWIN's trade-off plot of variance (V) and compression rate (R) of network radius with respect to window of aggregation ω.	69
14	Network radius time series for the Enron dataset at three levels of aggregation: (a) fine level of aggregation, $\omega = 1$ day, (c) coarse level of aggregation, $\omega = 12$ days, (b) the right level of aggregation, $\omega = 5$ days.	70
15	Network density for the Reality Mining dataset (a) and Haggle dataset (b) at three levels of aggregation: too fine level of aggregation(top picture), the right level of aggregation (middle picture) and too coarse level of aggregation (bottom picture).	71

LIST OF FIGURES (Continued)

FIGURE **PAGE**

16 TWIN's results for the measure of network density of the $DynUR$ stream. 72
17 TWIN's results for the measure of network density of the $DynMix$ stream. 73
18 Comparison of TWIN heuristic to Graphscope heuristic (a) and FFT method (b). 74
19 Average Path Length of original network (a), network perturbed within each partition (b), and network perturbed across partitions (c) for the Enron dataset. 77
20 Perturbation analysis of TWIN's performance for the Reality Mining and Haggle datasets. 78
21 Perturbation analysis of TWIN's performance for the $DynUR$ and the $DynMix$ datasets. 79
22 Average path length of original $DynMix$ stream(a), and oversampled stream(b) with oversampling factor $\alpha = 4$. 80
23 Clustering Coefficient of original Reality Mining stream (a), and oversampled stream (b) with oversampling factor $\alpha = 5$. 80
24 Two examples of intervals with same frequency, but different persistence pattern. 83
25 Illustration of the temporal bounds of the edge frequency function $freq$ for window size $\omega = 3$ and shift parameter $s = 1$. 85
26 Partitioning of the $DynMix$ stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line. 91
27 Partitioning of the Haggle stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line. 92
28 Partitioning of the Reality Mining stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line. 94
29 Illustration of the internal consistency and local monotonicity properties. 97
30 Illustration of Dynamic Link Prediction Problem 105

LIST OF TABLES

TABLE		PAGE
I	Results of TWIN heuristics for the Enron, Reality Mining, Haggle.	75

SUMMARY

Networks have become an indispensable data abstraction that captures the nature of a diverse list of complex systems, such as on-line social interactions, email and cell phone communications, or protein interactions in a cell. All these systems are inherently dynamic and change over time. The abstraction of choice for incorporating time has been the "dynamic network", a time series of graphs, each representing an aggregation of a small discrete time interval of the stream of interactions. While in many cases the system under observation naturally suggests the size of such a time interval, it is more often the case that the aggregation is arbitrary and is done for the convenience of the data representation and analysis. However, it is clear that the choice of the time interval at which the network is discretized and aggregated has great implications on the structures observed, analysis performed, and inference made about the nature of the network and the processes on it. **This thesis is the first to establish a framework for the problem of Temporal Scale Inference (TSI) for dynamic networks.** We formally define the TSI problem and explicitly present some of its associated challenges. We present an analytical framework for studying the characteristics of special cases of interaction streams as probabilistic processes. We give characterizations of a null model and define the notion of the "right" temporal scale of a list of structured interaction streams including the general class of oversampled, noisy stationary streams. We present an axiomatic framework that formalizes desired properties of the "right" temporal scale. This framework serves as a common ground for consistently

comparing the performance of different heuristics for the TSI problem. We present two heuristics for identification of the inherent temporal scale of interaction streams. Overall, this thesis focuses on the analysis of the scale of dynamic networks with the objective to make the "art of looking at the right scale" more scientific.

CHAPTER 1

INTRODUCTION

1.1 Overview

Complex systems arise in various domains such as sociology, biology and technology, to name but a few. These systems are often abstracted and analyzed as *networks*. Networks are graphs with nodes representing entities, such as people or proteins, and edges representing interactions between pairs of entities, such as, email communications between two people or participation of proteins in the same regulatory process. Complex systems are inherently dynamic and change in time. Their temporal dynamics are often analyzed by embedding the concept of network in time.

Whether it is on-line communications (12; 36; 35), animal social interactions (17; 21; 54; 62), or gene regulatory processes (27), the dynamic systems they represent have inherent rhythms at which they function. Some of these inherent rhythms come from the system itself, others are imposed by outside circumstances. Circadian patterns of cell regulatory systems, seasonality in mobility patterns of animals, daily and weekly communication patterns of humans are just a few examples of these characteristic temporal scales. Not only do these complex systems have inherent rhythms, different patterns within them form and live at different scales (34). For example, when analyzing animal population behavior three temporal scales are considered to be important for capturing the hierarchical nature of its social

structure (23): the scale of the interactions themselves, the scale of patterns of interactions (relationships), and, finally, the scale of patterns of relationships (network structures). In this context, grooming interactions of baboons usually have a temporal scale ranging from seconds to minutes, mother to infant or peer to peer relationships have a scale extending over years, while an individual troop membership, splitting or formation of new troops extends from years to decades (59). Similarly, in human social behavior, the patterns of interaction of conversations, friendships, and kinship occupy different temporal scales. Every dynamic complex system exhibits this kind of multi-scalar behavior.

We view the system through the filter of the data we collect. These data are typically collected opportunistically, with the temporal rate of data not always matching that of the system. In order to ask questions about these systems, the tools we use to answer the questions, and the temporal scale of analysis have to match the temporal scale of the process underlying the question. Whether the question of interest is the detection of anomalies, the understanding of cohesiveness and persistence of interactions, or the prediction of the system behavior, the temporal scale at which the analysis is applied needs to reflect the temporal scale that captures what is essential for the question. When we analyze millions of IP network traces in order to detect outlying behavior, should we analyze their communication patterns every five minutes, every hour, every day? How long should social interactions persist to be considered meaningful relationships in a social network (23; 46)? Just like the cell has the "capacity" to compute the temporal scope of mRNA expression (66), we would

like to develop an understanding of how to estimate the tempo of a given dynamic system by analyzing its expression as a series of interactions occurring in time (i.e. temporal stream).

1.1.1 The Dynamic Network Abstraction

The abstract representation of choice for modeling a dynamic system has been that of the *dynamic network*, also referred to as *temporal network* (25; 29; 31; 37; 38). A dynamic network is a time series of network snapshots. Each snapshot represents a state of the system over the interval of time such as a minute, a day, or a year in the life of the system. The duration of the snapshot represents the temporal scale of the dynamic network since all the interactions are lumped together discarding their order in time. Little thought, to date, has been given to matching this temporal scale to that of the system under study. A snapshot of a year is not appropriate for analyzing human conversation, but maybe right for understanding kinship relations, minute-long snapshots could be suitable for analysis of gene regulatory systems, but too fine for the baboon troop membership. **How, then, should we go about finding the "right" temporal scale for the dynamic network?** This is the central question of this thesis.

In the abstraction of the dynamic network, the temporal ordering of interactions within a snapshot is lost. All these interactions are represented as living in the same temporal scale, whether we have the finer temporal information or not. In some instances, the data already comes as a series of aggregated snapshots; in other instances, we are given a stream of interactions in time which we have to aggregate. We have to make sure that as we transition from the temporal interaction stream or a collection of finer snapshots to a

dynamic network representation, the information that we discard is not critical. It is not clear how to decide what the right temporal scale of dynamic networks should be. While in many cases the system under observation naturally suggests the size of such a temporal scale (37), it is more often the case that the choice of temporal scale is arbitrary and is done for the convenience of the data representation and analysis. For example, it is convenient and sometimes meaningful to analyze human interaction patterns in calendaric scales, but it does not always make sense to analyze animal social interactions in similar scales. Studies of periodic behavior of animals have shown that animals do not care much about week days and weekends (37).

Within the complex system there are subsets of interactions that form functional units that naturally co-occur together and their analysis as a cohesive unit allows us to see critical system behavior such as collective emergent behavior (46; 47; 52; 66). For example, when studying molecular mechanisms of diseases, it is important to study the interactions of relevant genes concurrently in order to observe their temporal coordination (66). Similarly, epidemiological studies show that analysis of concurrent relations allows for a more accurate estimation of the magnitude of spread of an infectious agent (47; 52). In all these scenarios, analysis of interactions as series of network snapshots allows us to uncover inherent concurrent sets of interactions while maintaing only the critical temporal orderings. In this transition from data streams to dynamic networks we have to know to discard the little noisy perturbations of functional units, while retaining the meaningful temporal ordering at the scale of natural functionality as a whole.

1.1.2 Empirical Motivation

The level of aggregation of the temporal stream has great implications on the patterns observed in the corresponding dynamic network and the inference made about the network and the processes on it (15; 25; 31; 34; 46; 60; 11). As Moody, McFarland, and Bender-deMoll (46) point out, if analysis is applied at too fine temporal resolution, we end up observing a network that has lots of temporal detail, yet the interesting and meaningful co-occurring patterns, such as communities, may not be fully formed. On the other end of the spectrum, when we aggregate the network at a too coarse of a temporal scale, we loose critical temporal information and cannot observe meaningful temporal changes to the system or processes over it. For example, ecologists Baldock et al. (2) have shown that analyzing plant pollinator interactions at a daily temporal scale misses temporal variations during the span of the day that are critical for correctly interpreting interactions as either competitive or facilitative. Figure 1(a) gives an illustration of the effect of the level of aggregation on the kind of network structures that we observe (image reprinted from (7), with authors' permission). Figure 1(b) also illustrates how measures computed over the dynamic network critically depend on the level of aggregation that generated this network (image reprinted from (15), with authors' permission).

1.1.3 Data Collection

Data about dynamic interaction systems is often collected as a sequence of interactions together with temporal information about their occurrences. Depending on the nature of the dynamic system, it might be more meaningful to represent the temporal dynamics as a

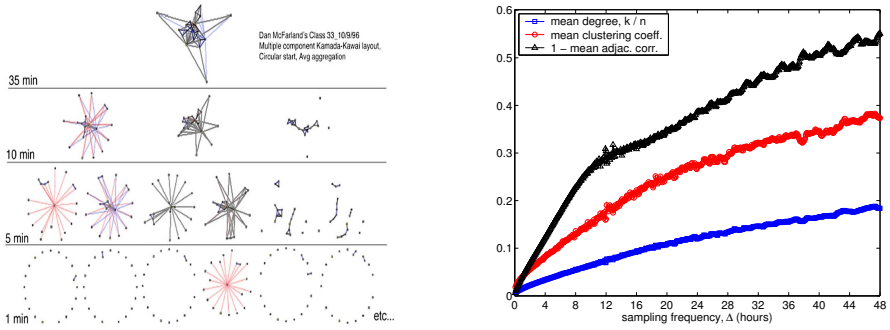
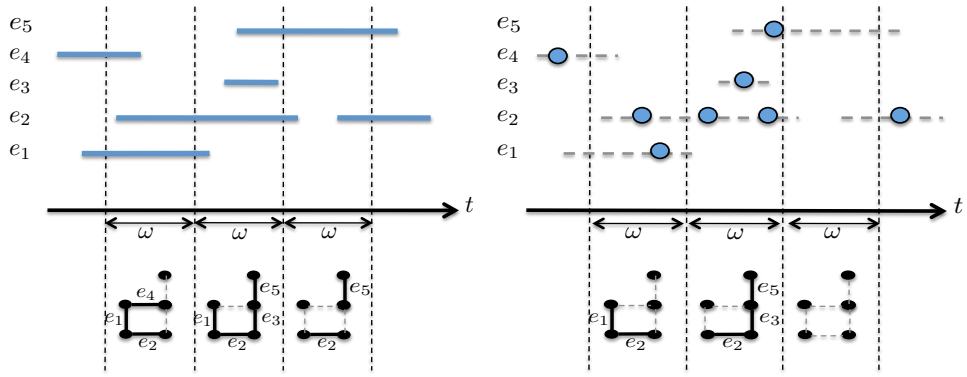


Figure 1: Representation of a social network aggregated at different windows of aggregation (a), and network measures of the Reality Mining dataset as functions of the window of aggregation (b).

stream of instantaneous interactions (i.e *point-based* interaction streams). At other times, a stream of interactions with temporal durations (i.e. *interval-based* interaction streams) is more suitable. For systems such as email communications, point-based streams offer a better representation. On the other hand, friendships in a social network or grooming in baboon troops are better characterized as interval-based streams. Interval-based interaction streams can be viewed as a generalization of point-based interaction streams, where we can think of the duration time as zero. In this thesis, we analyze both instances of interaction streams and show that their unique characteristics lead to different characterizations of the "right" temporal scale such as order invariance and persistence of structure. Figure 2 gives an illustration of both types of interaction streams.

Whether the collected data comes from GPS sensors, digital recording of emails, or human observation of animals grooming, most often, what we record is the instantaneous

times at which the interactions were observed to be present. This process often introduces different kinds of artificial noise that can be described both in terms of topological structure and in terms of temporal structure. Topological noise arises when we attempt to represent continuous behavior discretely (GPS). We might miss interactions that should be present in the network. At the same time, we might record interactions that occur spuriously, but are not meaningful. For example, when collecting proximity-based networks of students at MIT using bluetooth devices or zebras in Kenya using GPS collars, an interaction, then, is being at the same place at the same time for a sufficiently long interval. If two zebras happen to cross paths without actually interacting, devices that sample positions sufficiently frequently will record this occurrence even though this "interaction" should not be included in the network. On the other hand, observing baboon interactions once a day, will miss many important dynamics.



(a) Interval-based interactions.

(b) Point-based interactions.

Figure 2: Illustration of two different representations of time in interaction streams.

1.1.4 Oversampling of Temporal Streams

Oversampling is an aspect of the data collection process that can help with the issue of representing continuous time discretely. Oversampling helps reduce the number of missing or spurious interactions. It allows us to better understand what interactions are persistent in the network. On the other hand, oversampling affects our ability to distinguish between local noisy temporal orderings and critical temporal ordering. For example, when we observe human interactions at much higher rate than necessary, it is maybe hard to distinguish between interactions that develop independently of each other and those that are transient. When emails arrive within seconds of each other, is their ordering meaningful? Is it important in what order people walk in to a meeting room or is it more important to know that they were present at the meeting? With the advent of electronic data collection of interactions using communication devices, GPS, and proximity sensors, it is often the case that data are oversampled at orders of magnitude higher temporal resolution than the temporal scale of the underlying process. Therefore, it is important that the aggregation process correctly accounts for the oversampling effects.

So far we have discussed scenarios when the aggregation of interactions streams into a series of network snapshots is useful in capturing both the topological and temporal structure of the underlying system. The characteristic of the system that makes the dynamic network the right abstraction is the notion of local concurrency or temporal independence. In dynamic systems where the temporal ordering of the interactions is absolutely critical,

and there are no hidden concurrencies buried due to the data collection process, aggregation is not a useful tool.

In addition, we have discussed how temporal networks have an inherent rhythm that governs their dynamics. This is one natural way to define what is the right temporal scale. An alternative way would be to define the right temporal scale in terms of what is useful about networks. This leads to an orthogonal approach which is application driven. For example, the identification of the most frequent sub-graphs, or the identification of dynamic communities are useful applications that give us meaningful insight about the network. The natural question that arises here is to identify the optimal temporal scale at which application-specific patterns become detectable.

In summary, the dynamics of complex systems evolve at characteristic temporal scales. Understanding the mapping between structure and the temporal scale at which it lives is critical, and when done appropriately can lead to meaningful and insightful analysis. The subtle interplay between the temporal concurrency and temporal ordering is at the core of what is essential about the right temporal resolution for analysis of dynamic networks. The notions of temporal concurrency and temporal ordering depend on the context of analysis and they often lead to discovery of complex multi scalar nature of the structure. When temporal concurrency is something inherent to the system or analysis, aggregation of interactions streams helps in capturing this aspect of system functionality.

Although we have illustrated the challenges related to temporal scale identification in the context of dynamic complex systems, the problem of identifying the right resolution

for analysis of temporal data in general is very broad and covers many research areas. The relevant literature spans fields from signal processing (20; 50) and information theory (57) to time series analysis, time series segmentation (30; 49; 51), and model granularity (5; 22; 53). While the literature mentioned in this section offers a solid foundation for the problem of temporal scale identification in general, it does not explicitly address data that are represented as networks. It is not clear, for example, how techniques like aggregation or smoothing of numerical values relate to the same techniques applied to network structures (11; 43; 60). There is, however, the opportunity for great research in trying to translate and adapt these methods for the analysis of temporal scale of networks. Caceres, Berger-Wolf and Grossman (11) show that for special classes of network generative processes, the class of linear network measures (such as density and average degree) capture essential characteristics of the network at different scales, while Miller, Bliss and Wolfe (43) aim to develop a general signal processing theory for networks (graphs).

1.2 Main Contributions and Findings

The understanding and identification of the right temporal scale of dynamic networks is a nascent area of research. This thesis is the first to establish a framework for the problem of **T**emporal **S**cale **I**nference (TSI) for dynamic networks. The following is a more detailed list of contributions of this thesis:

- We formally define the **T**emporal **S**cale **I**nference (TSI) problem for dynamic networks and explicitly present some of its associated challenges.
- We present an analytical framework to study the TSI problem.

- We give characterization of a null model for the TSI problem.
- We define the notion of the "right" temporal scale of a list of structured interaction streams including the general class of oversampled, noisy stationary streams.
- We present an axiomatic framework that formalizes desired properties of the "right" temporal scale. This framework serves as a common framework for consistently comparing the performance of different heuristics for the TSI problem.
- We present two heuristics for identification of the inherent temporal scale of interaction stream.

1.3 Thesis Outline

The outline of the thesis is as follows:

Chapter 2 (Literature Review) We review existing literature related work to the TSI problem.

Chapter 3 (Problem Formulation for TSI) We present some basic definitions and descriptions of concepts used throughout the thesis, as well as formally define the TSI problem.

Chapter 4 (Special Cases of Interaction Streams) We study special cases of interaction streams and illustrate important characteristics of the notion of the right temporal scale. In addition, we present a formal analytical approach for studying the TSI problem.

Chapter 5 (Axiomatic Framework) We formally define a list of desired properties for the "right" temporal scale of interaction streams.

Chapter 6 (Inherent Timescale) We present two approaches for identification of the inherent temporal scale of interaction streams.

Chapter 7 (Future Work and Conclusions) We discuss directions for future work for the TSI problem and give concluding remarks.

1.4 List of Publications and Manuscripts

Some of the research presented in this thesis has appeared in or will appear in refereed conferences and journals. Here, we list the relevant publications associated with each chapter.

Chapter 4 (Special Cases of Interaction Streams)

- **R. S. Caceres**, T. Berger-Wolf and R. Grossman. Temporal scale of processes in dynamic networks. In ICDMW '11. Vancouver, CA. 2011. [Ref. (11)]

Chapter 5 (Axiomatic Framework)

- **R. Sulo**, T. Berger-Wolf. Temporal Scale for Dynamic Networks. *In P. Holme and H. Saramäki (Eds.). Temporal Networks.* Springer Complexity Series. 2012 (to appear)

Chapter 6.1 (Inherent Timescale)

- **R. Sulo**, T. Berger-Wolf and R. Grossman. Meaningful selection of temporal resolution for dynamic networks. In MLG '10. New York, NY. 2010. [Ref. (60)]

CHAPTER 2

RELATED WORK

Following is a brief review of related work in the areas of signal processing, information theory and time series analysis.

2.1 Signal Processing & Information Theory

Temporal aggregation is a natural pre-processing step when the frequency at which the data are generated is lower than the frequency at which the data are sampled. Usually the approach involves formulating a trade-off between loss of information and reduction of noise present in the signal. The goal in this context is to identify the window at which the original continuous signal can be fully recovered from the discretized signal. The Nyquist-Shannon sampling theorem (57) gives a necessary condition for the length of the sampling window in this context.

Minimum description length (MDL) is another information theoretic technique that is used to find the best granularity for data analysis (22; 53; 5). In this context, information is defined in terms of its algorithmic complexity. It assumes that the best hypothesis for a given set of data is the one that leads to the largest compression. Several works have used MDL principal to learn the best model granularity. Identifying the intrinsic temporal scale of time series can be viewed as a special case of model granularity and there are a few approaches that apply the MDL concept directly to such data (26; 63). In the next

section we review the approach by Sun et al. (61) that applies the MDL concept to dynamic networks.

2.2 Time Series Smoothing

Smoothing techniques are prevailing in the domain of time series analysis. In this context, some of the variation in the data is assumed to be due to random noise. The goal of the smoothing techniques is, therefore, to cancel some of the variations and to reveal inherent properties of the time series, such as trends or seasonal and cyclic behavior. The two main groups of smoothing methods are the averaging methods and the exponential smoothing methods.

While the literature mentioned in this chapter offers a solid foundation on how to properly aggregate data for analysis, it does not explicitly address datasets that are represented as networks and, furthermore, it does not address the dynamic nature of these networks. In our line of research, we focus explicitly on understanding how the process of aggregation or smoothing of interaction streams affects the quality of dynamic network that we get.

2.3 Temporal Scale of Dynamic Networks

The problem of identifying the right temporal scale for dynamic networks has only recently started getting the deserved attention. James Moody explicitly points out the problem in (46). Existing literature on the topic is preliminary and mainly of empirical nature. Clauset and Eagle (15) illustrate the effect of the aggregation window in understanding the periodic dynamics of the Reality Mining dataset (18). They recommend the use of Fourier Transform analysis and auto-correlation analysis of time series of network

measures. While these techniques have been successfully applied to understand stationary time series, their application to time series of measures originating from highly dynamic and complex networks might not be appropriate. It is not clear how the aggregation of time series of measures in networks relates to the underlying aggregation of interactions. Caceres et al. (11) theoretically show that for the special class of oversampled stationary processes and the special class of linear network measures (such as density and average degree), there is a direct relation between the two. We do not know, however, whether the same is true for more general dynamic network processes and network measures. Sulo, Berger-Wolf and Grossman (60) propose a heuristic that applies aggregation at the level of the dynamic network rather than at the level of time series of network measures in order to preserve as much of the network structure. They also give an explicit formulation of the optimal window of aggregation using the information theoretic framework.

The approach by Sun et al., (61) developed initially for the purpose of efficiently clustering dynamic networks consists of grouping similar network snapshots into one time interval using the Minimum Description Length principle. The idea of compressing the graph to maintain only the more relevant aspects is very promising and relevant for the problem of aggregating at the right time scale and is similar to the approach in (60). The contribution of the method by Sun et al., (61) is to use drastic changes in the time series of compression levels to segment the timeline of the temporal network. Rather than focusing explicitly on change detection, the goal of the research in (60) is more general, that is, identifying the

inherent temporal scale that governs the overall dynamics of the network, as well as the changes in that scale.

CHAPTER 3

PRELIMINARIES AND PROBLEM FORMULATION

In this section we present some basic definitions and descriptions of concepts used throughout this thesis.

3.1 Definitions and Notation

Let V be a set of vertices and E the set of edges defined over $V \times V$. For $\forall e_{ij} \in E, i, j \in V$ and $t \in [1, \ldots, T]$, the pair (e_{ij}, t) is the time labeled instance of e_{ij}.

Definition 3.1.1. A *temporal stream of edges* E_t is a sequence of edges ordered by their time labels:

$$E_t = \{(e_{ij}, t) | e_{ij} \in E, t \in [1, \ldots, T]\}$$

Let X_{ijt} be a random variable representing the existence of an edge e_{ij} in the stream E_t at time t:

$$X_{ijt} = \begin{cases} 1 & \text{if } (e_{ij}, t) \in E_t \\ 0 & \text{if } (e_{ij}, t) \notin E_t \end{cases}$$

Let $\mathcal{P}$ be a partition of the timeline $[1, \ldots, T]$:

$$\mathcal{P} = [t_0, t_1), [t_1, t_2), \ldots, [t_k, T]$$

As a special case, we consider the *uniform partition* $\mathcal{P}_\omega$, where each interval p_i has length ω:

$$\mathcal{P}_\omega = \{p_k\} \text{ s.t.} \forall p_k, |p_k| = \omega$$

Definition 3.1.2. A *dynamic graph* DG is a sequence of graphs defined over stream E_t and a fixed partition $\mathcal{P}$ of E_t:

$$DG : \langle (V, E_t^k) \rangle$$

$$DG = \langle G_1, G_2, \ldots, G_k, \ldots G_{|P|} \rangle$$

with $E_t^k = \{(e_{ij}, t) | e_{ij} \in E_t, t \in p_k\}$ and each G_k is associated with the kth interval p_k in $\mathcal{P}$.

We now define the operation of aggregation of a temporal stream of edges into the time series of graphs comprising a dynamic network. Given the temporal stream of edges E_t, and a fixed partition of the stream $\mathcal{P}$, we define the aggregation function that takes as an input the temporal stream of edges and the partition and outputs a time series of graphs.

Definition 3.1.3. An *aggregation function* A on a temporal stream E_t, and a fixed partition $\mathcal{P}$ is defined as:

$$A : E_t \times \mathcal{P} \rightarrow \langle (V, E^k) \rangle$$

$$A(E_t, \mathcal{P}) = DG$$

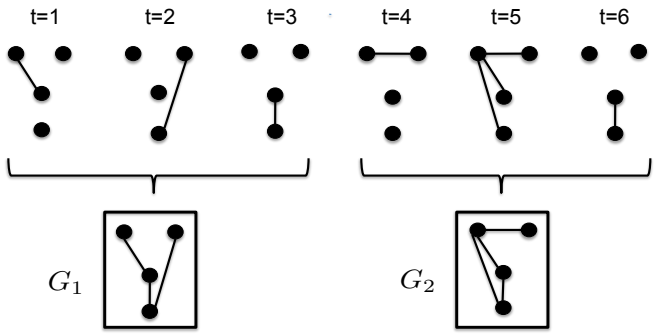


Figure 3: Aggregation of a temporal interaction stream with window of aggregation $\omega = 3$.

The aggregation function A takes all the edges occurring in a stream within a time interval $p_k \in \mathcal{P}$ and constructs a graph. Consider the scenario when A is applied to the uniform partition $\mathcal{P}_\omega$. Figure 3 shows an illustration of the aggregation function when the window of aggregation ω is 3. Note, that edges can occur within each temporal window more than once, but they get represented in the corresponding aggregated graph at most once. Throughout this thesis, we use this definition for the aggregation function. Another possible extension to this definition of A could take the multiplicity of edge occurrence into account. In a more general sense, an aggregation function could also use as a parameter a "goodness of fit" measure μ, that allows to map an interval $p_k \in \mathcal{P}$ to the best fit graph G_k^* with respect to μ:

$$A(E_t, \mathcal{P}, \mu) = \langle G_k^* \rangle$$

Note that in the general definition, G_k^* does not necessarily have to include all the edges that occur during p_k.

Let Q be a function that measures the quality of this dynamic graph:

$$Q : \mathcal{P} \times DG \rightarrow \mathbb{R}^{+}$$

Quality function Q maps the pair $(\mathcal{P}, DG)$ to the set of non-negative real numbers so that these values capture how "good" the dynamic network is. Q can also be used to compare different dynamic network representations of the same temporal stream. The notion of the quality function is different from that of the objective function for a particular algorithm that generates a dynamic network. Thus, the use of quality function as a model selection tool allows us to use it for comparing different algorithms on the same data.

Linear functions defined over the edges of a graph $G(V, E)$ are of particular interest when analyzing structural properties of the graph. Let f be such a function:

$$f : E \rightarrow \mathbb{R}^{+}$$

$$f = \sum_{i,j \in V} a_{ij} X_{ij},$$

where X_{ij} is the event of an edge e_{ij} being present in the graph. An example of a linear function is *density*, the proportion of the number of edges present in a graph relative to the possible number of edges $\binom{|V|}{2}$. In this case case, $a_{ij} = 1/\binom{|V|}{2}$ for all edges e_{ij}. Other graph measures on graphs, can be defined similarly and are of great interest when studying graphs that evolve in time.

The following is a list of definitions of other graph theoretic measures. Some of these measures represent more complex (nonlinear) functions on the edges of the graph.

Number of Connected Components: a connected component is a set of nodes mutually reachable by paths in the graph.

Size of Giant Component: the size of the largest connected component.

Geodesic between a pair of nodes: the path with the smallest number of edges.

Eccentricity of a node: the greatest geodesic between the node and any other node in the graph.

Diameter: the maximum eccentricity of any node in the graph.

Radius: the minimum eccentricity of any node in the graph.

Average Path Length: the length of the average geodesic between any pair of nodes.

Clustering Coefficient: the number of triangles over the number of possible triangles in the graph (48).

Clique Number: the size of the largest clique.

Spectral Gap: the difference between the first and the second eigenvalue of the Laplacian of the graph (14). *The Laplacian* of graph $G(V, E)$ is defined as

$$L = D - A$$

where $D = diag(d_1, ..., d_n)$ is the degree matrix of G and A is the adjacency matrix. Spectral gap of L is known to capture the connectivity properties of the graph (14) .

3.2 Problem Formulation

The abstraction and identification of the right temporal scale for transitioning from a dynamic stream of interactions into a meaningful and representative dynamic network is not a straight forward task. One natural way to define the "right" temporal scale of dynamic systems is as the scale of the inherent rhythm that governs their dynamics. Alternatively, one can argue that the definition of the temporal scale depends on the analysis objective for a given dynamic network. In either case, there is an implicit notion of a quality function that characterizes the optimal aggregation of the interaction stream. Ultimately the goal is to identify the temporal resolution that corresponds to either global or local optima of this quality function. With this in mind, we now formally define the Temporal Scale Inference problem:

Definition 3.2.1. Temporal Scale Inference (TSI) Problem: Given a temporal stream E_t and a quality function Q, find the partition $\mathcal{P}^*$ of the timeline $[1, \ldots, T]$, and the corresponding dynamic graph DG^*, that optimizes the quality function Q:

$$\langle \mathcal{P}^*, DG^* \rangle = \arg\max_{\mathcal{P}, DG} Q(\langle \mathcal{P}, DG \rangle).$$

Now that we have given the definitions and stated the problem, we are ready to discuss in more detail some results for the TSI problem.

CHAPTER 4

EXAMPLES AND SPECIAL CASES

We have been discussing in general the temporal scale for dynamic networks and the definition is not straightforward. In this chapter, we study a collection of interaction streams for which we have some intuitive understanding on what is the "right" temporal scale. As we carefully define their generative processes and the properties they inherit, the goal is to understand more rigorously, and gain insight into what happens as we aggregate the streams at different temporal scales. We discuss examples ranging from the very simple constant stream with no temporal scale, to realistic interaction streams coming from real-world data. Each one of the examples touches on different aspects of temporal scale and helps us formalize the notion the "right" temporal scale.

4.1 Constant Streams

Constant streams are the simplest possible temporal streams and one might even argue whether they are indeed temporal at all. They are streams where the same set of interactions occurs at each time step. The corresponding static graph SG is defined as follows:

Definition 4.1.1. *Static Graph (SG)* is the graph $G(V, E', p)$ defined over the set of nodes V and the set of edges $E' \subseteq V \times V$. Each edge in this graph occurs with probability $p = 1$ at any time t:

$$\forall (e_{ij}, t) \in E' \subseteq V \times V, \quad Pr[(e_{ij}, t) \in G)] = 1$$

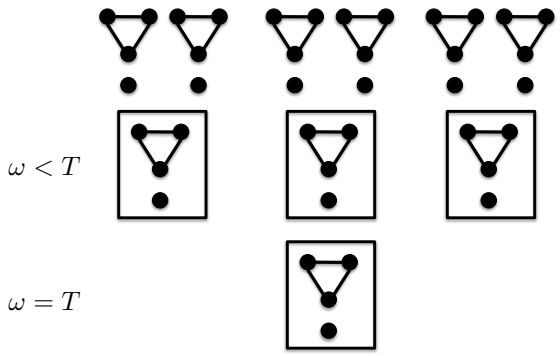


Figure 4: Aggregation of a static stream at different levels of aggregation.

Clearly there is no dependence on time and no algorithm should choose a particular temporal scale other than the entire time line. More explicitly, as illustrated in Figure 4, any aggregation of the constant stream over any aggregation window will produce a network identical to the original stream.

A more interesting case of a "stream with no temporal scale" is the stream where the set of interactions that appear at each step is not constant, yet, the occurrence of each interaction does not depend on time.

4.2 Random Unstructured Streams

4.2.1 *DynUR* Streams

We define the *Dynamic Uniform Random Graph* (*DynUR*) as the graph where each edge occurs at any time uniformly at random with probability p. This is the temporal equivalent of the Erdös-Rényi graph.

Definition 4.2.1. *Dynamic Uniform Random Graph (DynUR)* is the graph $G(V, E_t, p)$ with a constant probability $0 \leq p \leq 1$ for all edges

$$\forall (e_{ij}, t) \in \langle E, T \rangle \quad Pr[(e_{ij}, t) \in G)] = p.$$

Consider what happens when the aggregation function A (Definition 3.1.3) is applied to the $DynUR$ stream with aggregation window ω. The result is a time series of graphs, which we call $DynUR_\omega = A(DynUR, \mathcal{P}_\omega)$. Intuitively it is clear that the $DynUR_\omega$ graph is generated by a process with no temporal dependencies or correlations. The following Lemma shows that $DynUR_\omega$ is a time series of instances of the same Erdös-Rényi graph.

Lemma 4.2.1. *Every $G_k \in DynUR_\omega$ is a $G(|V|, q)$ Erdös-Rényi graph, where $q = 1 - (1 - p)^\omega$.*

Proof. By definition, $G_k \in DynUR$ is an Erdös-Rényi graph if each edge $e_{ijt} \in G_k$ exists with equal probability independently of other edges. The independence condition is trivially satisfied by the definition of the $DynUR$ graph. We now show that the $Pr[(e_{ij}, t) \in G_k]$ is also the same $\forall (e_{ij}, t) \in G_k$.

An edge (e_{ij}, t) exists in G_k, if it exists in at least one of the w time values representing the time window for G_k: $t \in [k\omega, (k+1)\omega)$. By definition, at any time t, $Pr[(e_{ij}, t) \in DynUR] = p$. Therefore, $Pr[(e_{ij}, t) \in G_k] = 1 - (1 - p)^\omega$. □

Note, that for $\omega = 1$, $DynUR_1$ represents the original stream of temporal edges with $t \in [1, ..., T]$ and each $G_k = G_t$ is a $G(|V|, p)$ Erdös-Rényi graph.

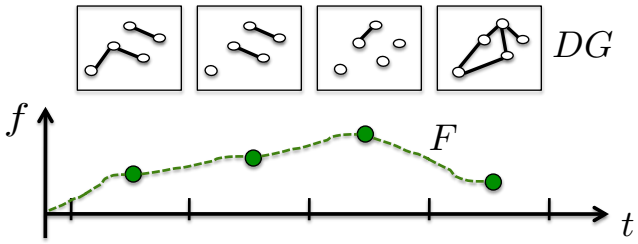


Figure 5: Time Series of a function f computed over a dynamic network.

4.2.1.1 Stationarity of functions on $DynUR$ Streams

Let f be a linear function on edges of a graph as described in Section 3.1. Let F be the resulting time-series that we get by applying function f to the dynamic graph $DynUR_\omega$ (illustrated in Figure 5). The following lemma shows that F is a covariance-stationary time series for any value of aggregation window ω:

Lemma 4.2.2. *$F(DynUR_\omega)$ is covariance-stationary. That is, for some constants μ, γ, and τ :*

(1) $\mathbb{E}_k[f(G_k)] = \mu$,

(2) $Cov(f(G_k), f(G_{k+\tau})) = \gamma_\tau$, $\forall 0 < k < T, \tau > 0$

Proof. Let G_k be a graph in $DynUR_\omega$. Let X_{ij}^k be the indicator variable for the event $(e_{ij}, t) \in \langle E_k, T\rangle$ for any $t \in [k\omega, (k+1)\omega)$. Then, $f(G_k) = \sum_{i,j\in V} a_{ij} X_{ij}^k$. Recall that while

edge e_{ij} can occur more than once in interval $[k\omega, (k+1)\omega)$, it shows up at most once in the aggregated graph G_k.

$$
\begin{aligned}
(1)\ \mathbb{E}[f(G_k)] &= \mathbb{E}[\sum_{i,j \in V} a_{ij} X_{ij}^k] \\
&= \sum_{i,j \in V} a_{ij} \mathbb{E}[X_{ij}^k], \text{ by linearity of expectation} \\
&= \sum_{i,j \in V} a_{ij} Pr(X_{ij}^k) \\
&= \sum_{i,j \in V} a_{ij} q, \text{ by Lemma 4.2.1} \\
&= \mu
\end{aligned}
$$

where μ is a constant with respect to the time index t.

(2) Let $X_{ij}^{k+\tau} = 1$ if $(e_{ij}, t) \in \langle E_{k+\tau}, T \rangle$ for any $t \in [(k+\tau)\omega, (k+\tau+1)\omega)$. By the definition of $DynUR$, $X_{ij}^k, X_{ij}^{k+\tau}$ are independent variables. $f(G_k) = \sum_{i,j \in V} a_{ij} X_{ij}^k$ and $f(G_{k+\tau}) = \sum_{i,j \in V} a_{ij} X_{ij}^{k+\tau}$ are independent variables as well, since they are defined as linear combinations of independent variables. Therefore, $Cov(f(G_k), f(G_{k+\tau})) = \mathbb{E}[f(G_k), f(G_{k+\tau})] - \mathbb{E}[f(G_k)]\mathbb{E}[f(G_{k+\tau})] = \mathbb{E}[f(G_k)]\mathbb{E}[f(G_{k+\tau})] - \mathbb{E}[f(G_k)]\mathbb{E}[f(G_{k+\tau})] = 0$

It is important to note that for the class of functions f, the property of stationarity is true at any value of aggregation of the $DynUR$ graph. □

4.2.1.2 Temporal Order Invariance of $DynUR$ Streams

Recall that the probabilistic permutation function π chooses at random a pair of edges $\langle (e_{i_1 j_1}, t_1), (e_{i_2 j_2}, t_2) \rangle$ from the temporal stream E_t and swaps their timestamps (Definition 5.1.1). Lemma 4.2.3 shows that reordering of edges according to function π has no effect on the outcome of aggregation of the $DynUR$ stream.

Lemma 4.2.3. *Aggregation of $DynUR_\omega$ is invariant under π :*

$$A(E_t, \mathcal{P}_\omega) = A(\pi(E_t, \mathcal{P}_\omega))$$

Proof. Let $\langle (e_{i_1 j_1}, t_1), (e_{i_2 j_2}, t_2) \rangle$ be the pair of edges chosen i.i.d. from E_t, by the permutation function π. We consider the two cases:

Case 1: $k\omega \leq t_1, t_2 < (k+1)\omega, 0 \leq k = \frac{T}{\omega} - 1$. By the definition of the aggregation function A, both edges $e_{i_1 j_1}, e_{i_2 j_2}$ belong to the same graph G_k, in the time-series $DynUR_\omega$. Therefore, even after π permutes the time labels of $e_{i_1 j_1}, e_{i_2 j_2}$, the aggregation function A will place them in the same graph G_k. Hence the resulting time series of G_k graphs will be identical before and after the permutation.

Case 2: $t_2 - t_1 > \omega$ We now compute the $Pr[(e_{ij}, t) \in G_k]$ after permutation. Lets consider the following mutually exclusive events:

1. Edge e_{ij} was selected by π, and it got swapped out from graph G_k.
2. Edge e_{ij} was not in graph G_k and it was not selected by π.

Let r=$Pr[e_{ij}$ selected by $\pi] = Pr[e_{ij}$ selected by $\pi | e_{ij}$ exists$] = \frac{p}{\binom{E_t}{2}}$. If edge e_{ij} was removed from G_k after it was selected by π, that means, edge e_{ij} occurred in exactly one time step during the time interval corresponding to G_k. Therefore, probability of event 1 happening is : $Pr[\text{event 1}] = rwp(1-p)^{w-1}$.

Let us now compute the probability of event 2. Probability of edge e_{ij} not occurring in G_K is $1-q$, where q represents the probability of edge e_{ij} being present in G_k by Lemma 4.2.1. Probability of edge e_{ij} not being selected by π is $1-r$. Therefore, probability of event 2 is: $Pr[\text{event 2}] = (1-q)(1-r)$.

The event of edge e_{ij} being present in G_k after the permutation is the complementary event of the union of event 1 and 2. Therefore, $Pr[(e_{ij}, t) \in G_k]$ after permutation is :

$Pr[(e_{ij}, t) \in G_k]$ after permutation =

$$= 1 - (Pr[\text{event 1}] + Pr[\text{event 2}])$$

$$= 1 - r\omega p(1-p)^{\omega-1} - (1-q)(1-r)$$

$$= 1 - r\omega p(1-p)^{\omega-1} - (1-p)^{\omega}(1-r)$$

$$= 1 - r\omega p(1-p)^{\omega-1} - (1-p)^{\omega} + (1-p)^{\omega} r. \quad (4.1)$$

□

The result from Lemma 4.2.3 shows that even though the permutation process changes the probability of an edge existing in each partition, the type of graph representing each partition is still an Erdös-Rényi graph, but more importantly, it is the same type of graph

across all the partitions. Also, note this result is true regardless of the value of the aggregation window, a unique characteristic of the $DynUR$ graph.

The results of Lemmas 4.2.1, 4.2.2, 4.2.3 all point to an inherent characteristic of the $DynUR$ stream, the uniformity of behavior across windows of aggregations. Whether we analyze $DynUR$ at the graph level as a series of Erdös-Rényi graphs, or at the function level by looking at the class of functions f, the behavior of $DynUR$ is invariant with respect to the window of aggregation. With this respect, $DynUR$ is an example of a more general class of interaction streams: *the streams with no temporal scale*. Even though we do not explicitly define all the generative processes that could lead to streams in this class, we believe that "the invariant behavior with respect to aggregation at any scale" is an important characterization.

Any stream with no temporal scale and in particular $DynUR$ can be looked as a representation of the null model for the TSI problem. In general, a null model provides a baseline and a sanity check for evaluating any algorithm claiming to solve a problem or assessing the significance of any discovered pattern. For static networks, the Erdös-Rényi graph has been used as the simplest null model (19). It is a simple graph where each edge occurs independently at random with the same probability p. This model generates a network where each vertex has the same expected degree. More sophisticated null models that can approximate the skewed degree distribution of empirical networks (4) have been proposed by Molloy and Reed (44; 45), and later on by Chung-Lu (13).

The definition of the null model becomes more complicated when we add time. In addition to correctly representing the relevant topological structure of the network, these models need to incorporate aspects of the temporal structure such as order, concurrency, and delay of interactions, among others. Null models for temporal networks have been proposed by Holme (24) and Karzai (28; 16). They use temporal reshuffling as a tool to generate streams where different aspects of temporal structure get randomized. Holme and Karzai discuss these null models in the context of analyzing processes, such as spread of a virus or information, over the temporal stream (to which they refer to as the contact sequence). Holme in (25) points to the issue that scale plays a role in capturing the critical temporal correlations:

> ... there are several kinds of possible temporal correlations and several time scales where the correlations are important, and thus no single, general-purpose null model can be designed (the temporal configuration model). Rather, by designing appropriate null models, one may switch off selected types of correlations in order to understand their contributions to the observed time-domain characteristics of the empirical temporal network.

In the absence of knowing the right temporal scale for the given temporal stream or the spread process over it, permutations at all the scales need to be tried and tested in order to generate the relevant null model. A null model is a way to test the importance of structure found at any given scale. It does not allow us, however, to find the temporal scale (or scales)

in and of itself. We can use a null model in conjunction with a quality function to do so. In Chapter 5 we formalize and generalize this observation for any quality function.

4.3 Structured Temporal Streams

4.3.1 Periodic Streams

We will now consider periodic streams for which we expect the period to be related to the concept of the "right" temporal scale. We give a general definition of the corresponding probabilistic graph, the *Dynamic Mixture* ($DynMix$) graph.

Definition 4.3.1. *Dynamic Mixture Graph ($DynMix_{M,\{w_l\}}$) on M fixed probability distributions ($DynMix_{M,\{w_l\}}$).* Given M probability distributions $\{P_l\}_{l=1}^{M}$ and a set of temporal windows $\{w_l\}_{l=1}^{M}$ such that $W = \sum_{l=1}^{M} w_l$, the *Dynamic Mixture Graph* $DynMix_{M,\{w_l\}}$ is the graph $G(V, E_t, P_l)$, such that

$$\forall (e_{ij}, t) \in E_t \qquad Pr[(e_{ij}, t) \in G)] = p_{lijt}$$

where

$$p_{lijt} = P_l(X_{ijt}) \quad \text{and} \quad l = \mathrm{mod}_M \left\lfloor \frac{t}{W} \right\rfloor .$$

The Dynamic Mixture Graph is a repeating sequence of Dynamic Graphs. Consider two special cases for the probability distribution functions generating the $DynMix_{M,\{w_l\}}$ graph:

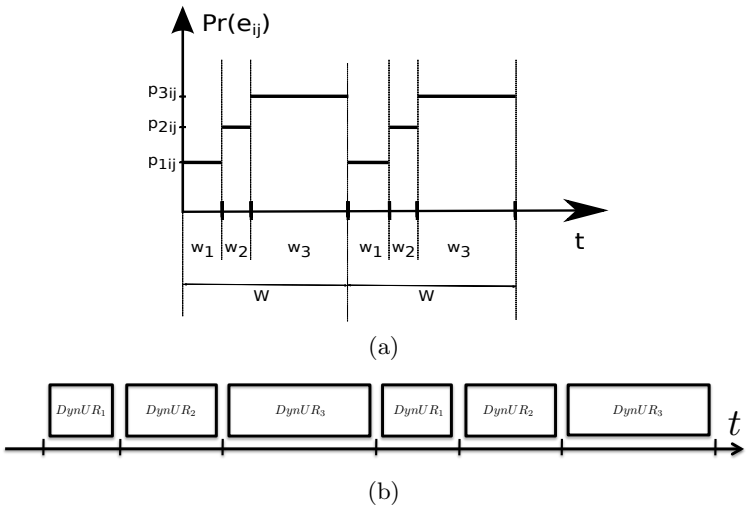


Figure 6: Illustration of two cases of the $DynMix$ graph with three alternating probability distributions (M=3). Subfigure (a) illustrates Case 1 of $DynMix$ with a cycle of constant probabilities for each edge, and Subfigure (b) illustrates Case 2 of $DynMix$ with a cycle of Erdös-Rényi graphs.

Case 1: A sequence of constant probability distribution P_l, so the probability of an edge e_{lij} does not depend on the time index t:

$$Pr[(e_{ij}, t) \in DynMix_{M,\{w_l\}}|(e_{ij})] = p_{lij}.$$

Figure 6 (a) gives an illustration of such a Dynamic Mixture Graph when the number of repeating probability distributions M is 3.

Case 2: A sequence of $DynUR$-s, where for any given probability distribution P_l, and a given time index t, the probability of all edges e_{lij} at t is the same:

$$Pr[(e_{ij}, t) \in DynMix_{M,\{w_l\}}|t] = p_t \quad \forall i, j \in V.$$

Figure 6 (b) gives an illustration of such a Dynamic Mixture Graph with M=3.

Note that $DynUR$ can be viewed as a special instance of both Case 1 and Case 2 of the $DynMix_{M,\{w_l\}}$ graph with $p_{lij} = p_t = p$ for all tuples $\{l, i, j, t\}$.

We will first consider Case 1 of the $DynMix_{M,\{w_l\}}$. Recall that in Case 1, probability of an edge e_{lij} does not depend on time index t, for any given probability distribution P_l: $Pr[(e_{ij}, t) \in DynMix_{M,\{w_l\}}|(e_{ij})] = p_{lij}$. We consider what happens when the aggregation function A is applied to a temporal stream of edges, each of them representing an edge in the $DynMix_{M,\{w_l\}}$ graph. The result of aggregating $DynMix_{M,\{w_l\}}$ is a time series of graphs which we call $DynMix_\omega = \mathcal{A}(DynMix_{M,\{w_l\}}, \omega)$. We will show that in the case of $DynMix_\omega$, stationarity of a linear function on edges is guaranteed only when the aggregation is done at the period level $W = \sum_{l=1}^{M} w_l$ (or any multiple of the period).

Lemma 4.3.1.

a) *The time series $F(DynMix_\omega)$ is covariance-stationary when the window of aggregation ω is a multiple of W, $\omega = nW$, where $W = \sum_{l=1}^{M} w_l$, and $n \in \mathbb{Z}$.*

b) *If $\omega \neq nW$, $\exists\omega$ s.t. $F(DynMix_\omega)$ is not covariance-stationary.*

Proof. a) Let $n = 1, \omega = W$. Let G_k be a graph in $DynMix_W$. Then, $f(G_k) = \sum_{i,j \in V} a_{ij} X_{ij}^k$, where X_{ij}^k is the indicator variable for the event $(e_{ij}, t) \in E_{k,t}\rangle$ for any $t \in [kW, (k+1)W)$. Therefore, $X_{ij}^k = 1$, if (e_{ij}, t) is generated from P_1, or

P_2, or ..., P_M. By this observation, the probability of an edge e_{ij} being in G_k is $Pr[X_{ij}^k = 1] = \sum_{l=1}^{M} p_{lij}$. Then the expectation of function $f(G_k)$ is:

$$\begin{aligned}
\mathbb{E}[f(G_k)] &= \mathbb{E}[\sum_{i,j \in V} a_{ij} X_{ij}^k] \\
&= \sum_{i,j \in V} a_{ij} \mathbb{E}[X_{ij}^k] \\
&= \sum_{i,j \in V} a_{ij} \sum_{l=1}^{M} p_{lij} \\
&= \mu, \text{ where } \mu \text{ doesn't depend on k.}
\end{aligned}$$

Note that the proof trivially generalizes to $\omega = nW$ for arbitrary values of $n \in \mathbb{Z}$. By the periodicity of $DynMix$, $X_{ij}^k = X_{ij}^{k+\tau}$, and furthermore $f(G_k) = f(G_{k+\tau}) = \sum_{i,j \in V} a_{ij} X_{ij}^{k+\tau}$. Therefore, $Cov(f(G_k), f(G_{k+\tau})) = Var(f(G_k))$

b) Now consider the case when $DynMix_\omega$ is aggregated at windows $\omega \neq nW$. We will show, by giving an explicit example, that there exists a window of aggregation ω at

which the time series $F(DynMix_\omega)$ is not stationary. For simplicity, assume $w_l = w$ for each P_l. Let the window of aggregation $\omega = w$.

$$\begin{aligned}
\mathbb{E}[f(G_k)] &= \mathbb{E}[\sum_{i,j \in V} a_{ij} X_{ij}^k] \\
&= \sum_{i,j \in V} a_{ij} \mathbb{E}[X_{ij}^k] \\
&= \sum_{i,j \in V} a_{ij} p_{lij} \\
&= \mu
\end{aligned}$$

Since the value of p_{lij} depends on the value of k, μ is not constant with respect to k. Therefore, the time series $F(DynMix_w)$ is not stationary. Similar results can follow for Case 2 of probability distribution functions $\{P_l\}$. Recall that in Case 2, each P_l is a function that is constant over the edges and only depends on time index t. Following similar arguments as in the proof of Lemma 4.3.1, it can be shown that time series F is stationary for any window of aggregation $\omega = nM, n \in \mathbb{Z}$, and that for values aggregation $\omega \neq nM$, there exist some ω when the corresponding time series F is not stationary.

□

We showed that for a periodic stream, a linear function on the corresponding dynamic graph becomes stationary at specific windows of aggregation. These windows of aggregation

correspond to the period (or any multiple of the period) of the underlying edge probability process.

4.3.2 Stationary Streams

One of the most intuitive properties that we want the dynamic network to have is stability or stationarity (34; 9). In physics this property is typically referred to as steady state, while in statistics they it is called stationarity. Whether we are trying to identify long term trends or typical behavior, or whether we want to predict new behavior, a stable system is a necessary condition for a meaningful analysis if we wish to infer something about the system from a history of observations. Furthermore, as perturbation analysis has increasingly become a powerful tool for untangling the complex structure of networks (28; 24), it is important to apply such analysis over a stable system. Otherwise, it is difficult to distinguish between changes due to the instability of the systems and changes due to the perturbation (9). Stability is a property analyzed extensively in the context of numerical time series. The interest in our work is to understand this property in the context of temporal interaction streams. Ultimately, the goal is to be able to identify aggregation levels (temporal scales) of the interaction stream so that the corresponding dynamic system represents a system in a steady state and, therefore, appropriate for analysis.

In this chapter, we will use the statistical definition of stability. More precisely, we define a stationary probabilistic function that generates the temporal stream.

Let P_t be the general case of a (weak) stationary probability distribution function generating the stream of edges $E_t = \{(e_{ij}, t), e_{ij} \in E, t \in [1, ..., T]\}$. That is,

1. $\mathbb{E}[p_{ijt}] = \mu_{ij}$, s.t. μ_{ij} does not depend on t.

2. $Cov(p_{ijt}, p_{ij(t+\tau)}) = \gamma_{ij}$, s.t. γ_{ij} does not depend on t.

4.3.3 Theseus Ship Streams

So far we have discussed streams whose structure is stable over time. We now turn our attention to a stream that appears to be stable, yet it changes slowly over time. What kind of issues does a stream like this introduce to our problem of scale? The notion of an object changing slowly over time is an old one, and is illustrated by the Theseus Ship paradox (56).

> The ship wherein Theseus and the the youth of Athens returned [from Crete] had thirty oars, and was preserved by the Athenians down even to the time of Demetrius Phalereus, for they took away the old planks as they decayed, putting in new and stronger timber in their place, insomuch that this ship became a standing example among philosophers, for the logical question of things that grow; one side holding that the ship remained the same, and the other contending that is not the same. [Plutarch, Life of Theseus]

The Theseus Ship paradox raises the question of whether the identity of an object fundamentally changes when all of the object's components have changed. Many great minds from ancient Greece to the present have struggled to find the right answer to the dilemma posed by this paradox. A translation of the Theseus Ship paradox in the framework

of dynamic networks has been discussed in (64) to describe the notion of communities whose members change over time.

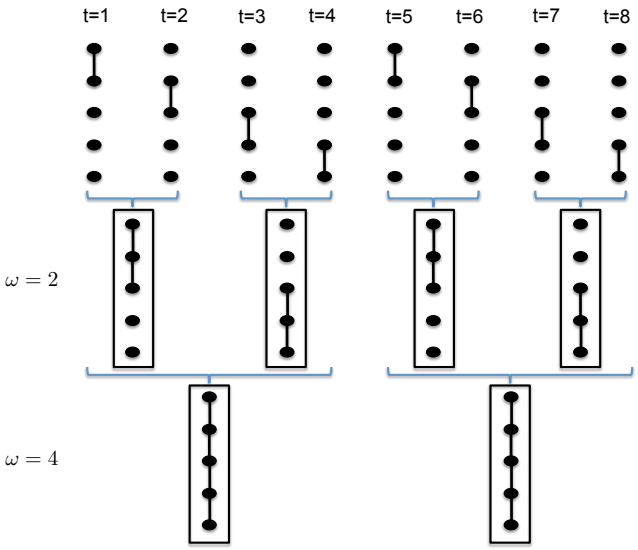


Figure 7: Aggregation of Theseus Ship stream at different scales. The original Theseus Ship stream has no interactions as the base set and one interaction changes every time step.

There is an analog of this paradox in the context of temporal interaction streams. Consider a process, where every k time steps, the same set of interactions occurs, except for a small change; one of the occurring interactions is replaced with a new interaction (illustrated in Figure 7). After enough time steps, the initial set of interactions is replaced completely with a new set of interactions. The question that arises here is "What is the right temporal scale for aggregating such a stream?" At a fine temporal scale we observe change that is

too gradual. At the same time, we are able to capture the persistence structure of the network. At a coarser temporal scale, we will loose this persistence structure, but we will be able to identify periodicity. This is a good example of the complexity of the definition of the "right" temporal scale. The dichotomy of persistence versus periodicity motivates the position that the definition of the "right" temporal scale is context- and question-specific. Furthermore, the aggregation the Theseus Ship stream illustrates the multi scalar nature property of temporal streams. Depending on the magnitude of change we want to observe in such a stream, different levels of aggregation are suitable for the analysis.

4.3.4 Oversampled Streams

Another property of the streams that is critical for the TSI problem is the oversampling property. The assumption that the process is oversampled is a natural one for any good data set. An under-sampled process cannot be guaranteed to contain sufficient information for analysis, by definition. Furthermore, given the pervasiveness of fast and automated data collection systems, oversampling is more of a realistic property rather than a wishful one. It does, however, introduce some unwanted side effects, such as artificial time orderings and spurious patterns. Naturally, we are interested in identifying aggregation levels of the stream that take the oversampling factor into consideration. At such temporal scale, the oversampling noise has been smoothed out, and the corresponding dynamic network is a true representative of the underlying dynamics. Specifically, if we have an existing stream where we know the optimal window of aggregation is ω, then, intuitively, if we over-sample by a factor of α, the new optimal window of aggregation should be $\alpha\omega$. Although the size of

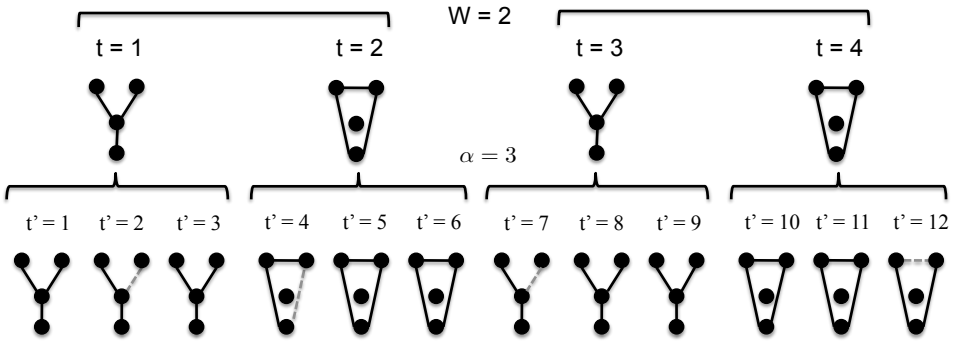


Figure 8: Oversampled periodic stream.

the window of aggregation changes proportionally to the oversampling factor, the process of finding the optimal window should not be sensitive to the oversampling factor. In this sense, uniformly modifying the frequency of interactions should not affect the relative temporal distances between interactions.

We now formally define the process of oversampling of interaction streams. Let P_t be the probability distribution function generating the stream of edges $E_t = \{(e_{ij}, t), e_{ij} \in E, t \in [1, \ldots, T]\}$.

Definition 4.3.2. An α-streching mapping ϕ_α of the time line $[1, \ldots, t, \ldots, T]$, where $\alpha > 0$, is defined as follows:

$$\phi_\alpha : [1, \ldots, t, \ldots, T] \rightarrow [1, \ldots, t', \ldots, \alpha T]$$

$$t' = [t\alpha - (\alpha - 1), t\alpha - (\alpha - 2), \ldots, t\alpha]$$

Definition 4.3.3. An oversampled probabilistic interaction process $P_{t'}$, over the time sequence $t' \in [1, \ldots, \alpha T]$, and probability function P_t, is defined as follows:

$$P_{t'} = P_{\phi_\alpha(t)} = \frac{1}{\alpha} P_t$$

An illustration of a simple oversampled periodic stream is given in Figure 8. Note that an oversampled periodic stream is still periodic (in this case period $W = 2$ and oversampling factor is $\alpha = 3$). The right scale for such a process takes into account the oversampling rate and recaptures what is essential about this process, whether that is the alternating change ($\omega^* = \alpha \frac{W}{2} = 3$) or the stationarity ($\omega^* = \alpha W = 6$).

As an additional example, consider the oversampled version of the $DynUR$ stream, we expect the properties of such a stream under aggregation to be invariant to oversampling.

Corollary 4.3.1. *Aggregation of $DynUR_\omega$ is invariant under the oversampling process:*

$$A(E_t, \mathcal{P}_\omega) = A(E_{\phi_\alpha(t)}, \mathcal{P}_\omega)$$

Proof. Based on Definition 4.3.3, the probability of each edge in the oversampled $DynUR$ stream $E_{t'}$ would now be $\frac{p}{\alpha}$, and Lemma 4.2.2 (stationarity of linear functions at any scale) will be trivially satisfied. □

Throughout this work, we are assuming oversampling at a uniform rate. In reality, data and the processes they represent are messy and oversampling could happen at nonuniform

rates. Although, we do not explicitly address this scenario in the thesis, analysis of streams at nonuniform oversampling rate presents an important direction for future work.

4.3.5 Noisy Streams

Similar to the definition of structure in dynamic networks, noise comes in two flavors: topological and temporal. Topological noise has to do with the presence or absence of an observed interaction (or a set of interactions) that does not reflect the behavior of the underlying system. The addition of time adds to the complexity of noise in dynamic networks. Specifically, the occurrence time of an interaction could be noisy as well as the ordering of this occurrence time with respect to that of other interactions. Alternatively, we can view noise as the antithesis of structure. In this context, we discussed in Section 4.2.1 the charactistics for the $DynUR$ stream, the completely structureless stream.

In general, the topological and temporal aspects of noise are coupled in ways that make it difficult to analyze them individually. Yet, throughout Chapter 4, we attempt to state the characteristics of temporal noise more explicitly. In Section 4.3.6, we analyze a simple case of temporal noise, by introducing gaussian noise to the temporal probabilities of each interaction. In addition, throughout the thesis, we use perturbation analysis as a way to detect noisy temporal orderings of interactions. Clearly, in real-world datasets, noise is generated by much more complex models and can have other manifestations that we do not consider in this thesis. In addition, we would expect the magnitude of noise to have an effect on the "right" level of aggregation. We would expect that, at some threshold, the large magnitude of noise overwhelms structure and the stream essentially becomes the $DynUR$

stream. Despite the fact that we do not address here these aspects of noise explicitly, they provide interesting and important directions for future work.

4.3.6 Oversampled, Noisy Stationary Streams

A periodic process is just one example of a stationary process. Now that we have also discussed oversampled and noisy streams individually, we can discuss a much more general class of interaction streams: the *oversampled noisy stationary* streams.

Let P_t be the general case of a (weak) stationary probability distribution function generating the stream of edges $E_t = \{(e_{ij}, t), e_{ij} \in E, t \in [1, \ldots, T]\}$. An oversampled noisy probabilistic process $P_{t'}$ defined over time sequence $[1, \ldots, t', \ldots, \alpha T]$, and probability function P_t is defined as follows:

$$P_{t'} = \frac{1}{\alpha} P_t + \epsilon,$$

with $\epsilon \in N(0, \sigma)$ representing Gaussian noise. Let $E_{t'}$ be the stream of edges generated by $P_{t'}$. Note that if $Pr[(e_{ij}, t) \in E_t] = p_{ijt}$, then $Pr[(e_{ij}, t') \in E_{t'}] = p_{ijt'} = \frac{p_{ijt}}{\alpha} + \epsilon$. Therefore,

1. $\mathbb{E}[p_{ijt'}] = \mathbb{E}[\frac{p_{ijt}}{\alpha} + \epsilon] = \frac{\mu_{ij}}{\alpha}$,
2. $Cov(p_{ijt'}, p_{ij(t'+\tau)}) = \mathbb{E}[p_{ijt'} \times p_{ij(t'+\tau)}] - \mathbb{E}[p_{ijt'}]\mathbb{E}[p_{ij(t'+\tau)}] = 0$, because $p_{ijt'}, p_{ij(t'+\tau)}$ are independent variables.

Let $DG_{P_{t'},\omega}$ be the dynamic graph defined over $E_{t'}$ at window of aggregation ω :

$$DG_{P_{t'},\omega} = A(E_{t'}, \mathcal{P}_\omega)$$

$$DG_{P_{t'},\omega} = \langle G_0, G_1, ..., G_k, ...G_{\alpha T/\omega - 1} \rangle$$

Let F be the resulting time-series that we get by applying function f on $DG_{P_{t'},\alpha}$:

$$F = \langle f(G_0), f(G_1), ..., f(G_k), ...f(G_{\alpha T/\omega - 1})\rangle$$

In Theorem 4.3.6 we show that the time series of linear functions on the dynamic network corresponding to an oversampled, noisy stationary stream, become stationary only for particular windows of aggregation. Furthermore, we show that there are windows of aggregation that are not suitable to capture the stationarity of the stream.

Theorem 4.3.6. *Let $DG_{P_{t'},\omega}$ be a dynamic graph which is the result of aggregation over a window ω of a stream of edges, generated by a noisy covariance-stationary process oversampled at a rate of α. Let $F(DG)$ be the time series of a linear function over the edges of $DG_{P_{t'},\omega}$. Then:*

a) *$F(DG_{P_{t'},\omega})$ is covariance-stationary when the window of aggregation ω is a multiple of α;*

b) *There exists ω which is not a multiple of α, s.t. $F(DG_{P_{t'},\omega})$ is not covariance-stationary.*

Proof. **a)** Let $\omega = \alpha, (n = 1)$

$$
\begin{aligned}
(1)\ \mathbb{E}[f(G_k)] &= \sum_k f(G_k) Pr[f(G_k)]\ , k \in [0, \ldots, T-1] \\
&= \sum_k f(G_k) Pr[\sum_{t'=k\alpha}^{(k+1)\alpha} \sum_{i,j \in V} a_{ijt'} X_{ijt'}], && \text{by the definition of } f, G_k, \\
&= \sum_k f(G_k) \sum_{t'=k\alpha}^{(k+1)\alpha} \sum_{i,j \in V} a_{ijt'} Pr[X_{ijt'}] \\
&= \sum_k f(G_k) \sum_{t'=k\alpha}^{(k+1)\alpha} \sum_{i,j \in V} a_{ijt'} P_{t'}(X_{ijt'}) \\
&= \sum_k f(G_k) \alpha \sum_{i,j \in V} a_{ijt} (\frac{p_{ijt}}{\alpha} + \epsilon), && \text{where } t = \lfloor \tfrac{t'}{\alpha} \rfloor \\
&= \sum_k f(G_k) \sum_{i,j \in V} a_{ijk\alpha} (p_{ijk\alpha} + \alpha\epsilon), && \text{where } t = k\alpha \\
&= \mu_{P_{t'},\alpha}, \text{a constant with respect to } t'.
\end{aligned}
$$

The proof generalizes trivially to $\omega = n\alpha$, for arbitrary values of $n \in \mathbb{Z}$

(2) $Cov(f(G_k), f(G_{k'+\tau)}) = 0, \tau > 0$, follows by similar arguments used in the proof of Lemma 4.3.1.

(b) Let $\omega \neq n\alpha$. Consider the simple case of an underlying periodic process generating the temporal stream of edges. Let the period be α. Then, by Lemma 4.3.1, there exists a window of aggregation $\omega < \alpha$ such that F is not covariance-stationary.

□

4.3.7 Real-World Streams

The following is a list of real-world datasets that represent temporal streams observed in different context.

Enron Email is a publicly available dataset of e-mails sent between employees of the Enron corporation (58). Each email address represents a vertex and an email exchange represents an edge. Timestamps were extracted from message headers for each day of e-mail activities. We are using a cleaner version of the dataset covering email exchanges from October 1998 to February 2003.

Reality Mining network consists of social interactions among 90 MIT students and faculty over a nine month period (18). The dataset is designed based on the idea that spatial proximity between people implies a social interaction. Participants are equipped with Nokia 6600 smart phones and an edge between two participants exists if a blue tooth connection is recorded. The original quantization time step is four hours.

Haggle Infocomm network consists of social interactions among attendees at an IEEE Infocom conference (55). There were 41 participants and the duration of the conference was 4 days. The original quantization step is 10 minutes.

4.4 Experimental Results

In this section, we investigate empirically how the window of aggregation of interaction streams affects the kind of structures we observe on the corresponding dynamic network. In particular, we analyze the behavior of linear functions on edges of the dynamic network.

The goal is to empirically identify the difference in behavior of these functions when they measure processes that have temporal structure and those that do not. We use density and the average degree as examples of popular network measures that can be naturally expressed as linear functions on the edges of the network. We then analyze the time series of density and average degree as functions of the window of aggregations.

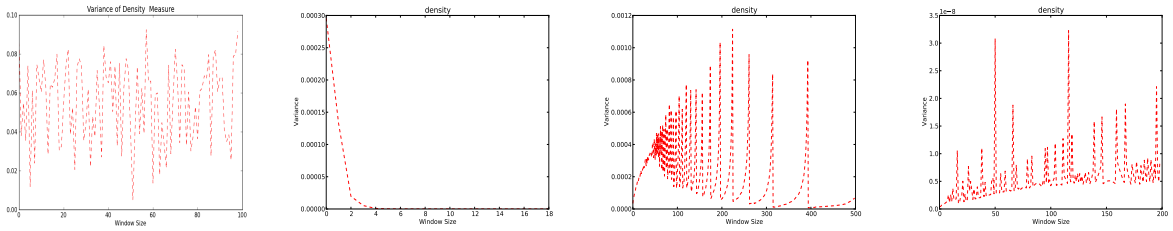
While there are several sophisticated methods to determine the stationarity of a time series, there is no standard, agreed process to do this. Often times, existing methods require artful tweaking of parameters. Since the goal in this work is not to determine stationarity per se, but to establish the differences between the different kind of streams, we use the variance of the time series as a simple proxy for stationarity. The following section gives a brief description of two simulated dynamic networks used for analysis:

DynUR **network** is a simulated dynamic network where each edge is present at any time t with some fixed probability p.

DynMix **network** is a simulated oversampled dynamic network, containing edges generated by two alternating fixed probability distribution functions. For our simulations, we used the Beta prime and the Gaussian distributions, and oversampling factor $\alpha = 5$.

The descriptions of the rest of the datasets used for the experimental analysis were described in detail in Section 4.3.7.

Figure 9 shows the plot of the variance of density as a function of the window of aggregation for each of the datasets mentioned above. The most immediate result illustrated



(a) $DynUR$ (b) $DynMix$ (c) Reality Mining (d) Haggle

Figure 9: Variance of the network density measure as a function of the window of aggregation for the $DynUR$, $DynMix$, Reality Mining and Haggle datasets.

in these plots is the fact that the variance function behaves distinctively different in the network with no temporal scale ($DynUR$), and the structured networks ($DynMix$, Haggle, and Reality Mining). The variance function is almost constant when computed over the $DynUR$ network (Figure 9 (a)). On the contrary, in the case of the structured networks, the variance function stabilizes only for particular windows of aggregation and there is a visible trend. As illustrated in Figure 9 (b), the variance of the density for the $DynMix$ network becomes stationary at window of aggregation 5 (and higher). This value corresponds to the oversampling factor used for simulating the $DynMix$ network. In the case of the Reality Mining and Haggle networks (Figure 9 (c), (d)) there is a correspondence between the periodicity of the dataset (1 day for Reality Mining and 30 minutes for Haggle), and the times when the variance function either approaches 0 or stabilizes. The behavior of the average degree function is almost identical to the behavior of density across all the datasets analyzed above.

4.5 Analytical Framework for the TSI problem

Throughout Chapter 4, we have used a consistent analytical framework for analyzing different classes of temporal interaction streams. Figure 10 gives a pictorial summary of this framework. We have presented different instantiations for each component in the framework. For example, we have given explicit formal definitions for special cases of generative probabilistic processes and their corresponding temporal streams. In addition, we have defined one version of the aggregation function A (Definition 3.1.3) that converts the temporal stream into a dynamic network.

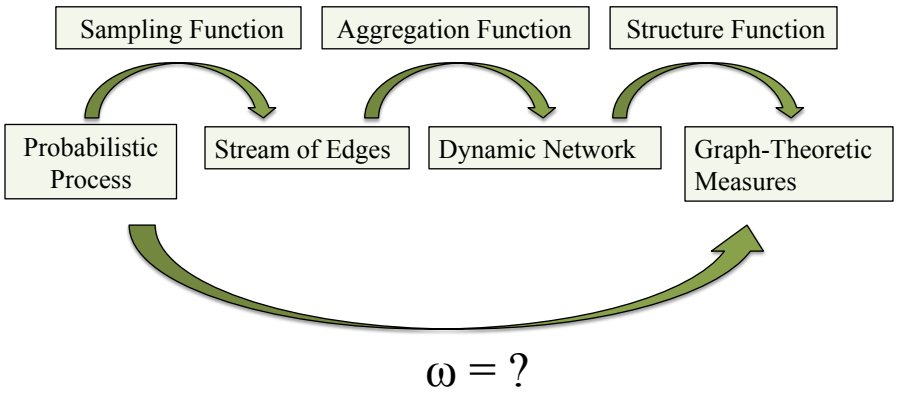


Figure 10: Analytical framework for analyzing the temporal scale of interaction streams.

This analytical framework led to some useful insights about the formal definition(s) of the "right" temporal scale for dynamic network. Detailed abstractions and applications of

these insights are discussed in detail through Chapters 5-6. Furthermore, this analytical framework can be easily extended to include definitions of other temporal streams, aggregation functions or network structural functions. The formal language that we have set up can be used to analyze more complex scenarios of temporal interaction streams, while maintaining a principled approach to the TSI problem.

CHAPTER 5

AXIOMATIC FRAMEWORK

In Chapter 4, we demonstrated some intuitive properties one would expect to observe for temporal streams. We showed that for a stream with no temporal scale, the re-ordering of interactions along the time line had no effect on the resulting dynamic network. We described some simple cases of interaction streams (e.g. $DynUR$ and constant stream) for which we have a clear understanding of the right temporal scale. Finally, we illustrated the effect of oversampling on the aggregation of different temporal streams and how it intuitively relates to the notion of the "right" temporal scale.

We now propose an axiomatic approach to capture this collection of insights in a formal way and to allow future rigorous analysis of the TSI problem. The axiomatic approach has recently gained a lot of interest in the field of spatial clustering. Similar to the TSI problem, the goal of spatial clustering is to postulate important sets of properties both in terms of the optimal partitioning (33), and in terms of the qualitative functions over such partitions (1; 6). The axiomatic view has also been applied to the analysis of graphs in the context of graph clustering (42) and graph complexity (10).

The TSI problem shares many of the characteristics and challenges of the clustering problem, both in metric and non-metric space, yet an axiomatic framework that synthesizes the characteristics of the TSI problem is lacking.

5.1 Axiomatic Framework: desired properties of the quality function Q

In the formulation of the TSI problem in Section 3.2, we defined the optimal temporal scale through the proxy of the quality function. This shifts the burden from finding the "real" temporal scale to optimizing the quality function. Ideally the two are the same. *What properties, then, should the quality function possess in order to correctly reflect the behavior of the underlying temporal scale?*

Here we present a set of axioms that delineate desired properties of the quality function Q. Let Ω be the set of all temporal streams E_t defined over the fixed set of vertices V and the finite time line $[0, \ldots, T]$. Since V is a fixed set and $[0, \ldots, T]$ is finite, there is a finite number of temporal streams in Ω. Therefore, Ω is a discrete probability space and we can define any suitable probability over it. Let $\mathcal{A}$ be a TSI algorithm, that takes as input a temporal stream E_t and outputs a set of pairs $\langle \mathcal{P}, DG \rangle$ as solutions:

$$\mathcal{A} : \{V \times V, [1, \ldots, T]\} \rightarrow \{\langle \mathcal{P}, DG \rangle\}$$

We can then think of Q as a random variable defined over the set $\{\langle \mathcal{P}, DG \rangle\}$. The range of Q is $[0, Q^*]$, where Q^* is the maximal value of quality associated with a specific partition $\mathcal{P}$ of E_t, and the corresponding dynamic network DG (here we think of $\mathcal{P}$ and DG as being equivalent to the probability space Ω):

$$Q : \Omega \rightarrow \mathbb{R}^+$$

Let Φ be a transformation function on the temporal stream E_t:

$$\Phi : E_t \to E_t.$$

An example of a transformation function is the permutation function π defined as follows:

Definition 5.1.1. Given a temporal stream E_t, a fixed partition $\mathcal{P}$ of the stream, a *probabilistic permutation function* π picks a pair of edges $\langle (e_{i_1 j_1}, t_l), (e_{i_2 j_2}, t_k) \rangle$, at random from E_t and swaps their timestamps:

$$\pi : E_t \to E_t$$

$$\langle (e_{i_1 j_1}, t_l), (e_{i_2 j_2}, t_k) \rangle \to \langle (e_{i_1 j_1}, t_k), (e_{i_2 j_2}, t_l) \rangle$$

We consider two special cases of function π:

- The *within-interval* permutation function π_w chooses the pair of edges from the same interval $p_i \in \mathcal{P}$ such that $t_l, t_k \in p_i$.
- The *across-interval* permutation function π_a chooses the pair of edges from two different intervals $p_i, p_j \in \mathcal{P}$ such that $t_l \in p_i$ and $t_k \in p_j$.

The goal of the following axioms will be to characterize the change in the quality of dynamic network DG defined over a given temporal stream E_t due to transformation function Φ. Let $\epsilon \geq 0$ represent a fixed threshold parameter characterizing the amount of change in quality. If the change is less or equal ϵ, we consider the change "small". Let $\delta \geq 0$ be a

confidence parameter about the probability of the change in Q being small (δ may depend on the size of the problem ($|V|$ and T)). We are now ready to formally define the axioms.

[Q1] Within Interval Order Invariance: *For an optimal partition, permutations of interactions within the same interval do not drastically change the quality of the dynamic graph.*

Formally, let $\mathcal{P}^*, DG^*$ be the optimal (with respect to a particular quality function Q) partition and the optimal dynamic graph for the temporal stream E_t. Let $\mathcal{P}^{*'}, DG^{*'}$ be the optimal partition and optimal dynamic graph corresponding to the perturbed stream $E'_t = \pi_w(E_t, \mathcal{P}^*)$. Then, with high probability, the change in the quality function after the perturbation is small:

$$\forall \pi_w, \forall p_i \in \mathcal{P}^*, P(|Q(DG^*) - Q(DG^{*'})| \leq \epsilon) \geq 1 - \delta.$$

Intuitively, the process of aggregating temporal interactions into a dynamic graph has the effect of assigning the same time to interactions within each partition, while preserving the temporal ordering across partitions. At the optimal temporal scale, the temporal ordering of interactions that fall within the same partition is not essential. The fact that, locally, some interactions are observed happening in a particular order is an artifact of looking at them at too fine of a temporal resolution. They could have happened in any order as long as they happened within a certain time frame.

[Q2] Across Interval Order Criticality: *For an optimal partition, permutations of edges across different intervals change the quality of the partition.*

Formally, we define the neighborhood of a given interval $p_i \in \mathcal{P}$ the following way:

$$N(p_i) = \{p_j | \ |i - j| \leq r, r > 0\}.$$

Let $DG^{*'}$ be the optimal dynamic graph corresponding to $E'_t = \pi_a(E_t, \mathcal{P}^*)$:

Then, with high probability, the change of the quality function after the permutation is substantial:

$$\exists \pi_a, \forall p_i \in \mathcal{P}^*, p_j \in N(p_i), P(|Q(DG^*) - Q(DG^{*'})| > \epsilon) \geq 1 - \delta.$$

Intuitively, at the optimal temporal scale, the temporal ordering of sets of interactions across the partitions is crucial and reflects the time dependence of the network structures. While this might not be true for all edges, there must exist a subset of interactions for which the ordering is critical. Otherwise, time does not play a role in the structure of the interactions.

[Q3] Measure Unit Invariance: *Uniform scaling of the oversampling factor does not change the quality of the dynamic network.*

Formally, let E_t and E'_t be two temporal streams generated by oversampling the same underlying process at rates α and α', such that $\alpha \neq \alpha'$. Let DG^* and $DG^{*'}$ be the

optimal dynamic graphs for E_t and E'_t respectively. Then, with high probability, the change in the quality function is small:

$$P(|Q(DG^*) - Q(DG^{*'})| \leq \epsilon) \geq \delta.$$

This axiom bears resemblance to the scale invariance axiom in spatial clustering. In this context, the partitioning of the data into clusters does not depend on the units of the distance function. The oversampling invariance axiom represents an analogous intuition: oversampling rate is a measure of the time unit used to measure how far apart interactions occur along the timeline. In this sense, uniformly modifying the frequency of interactions should not affect relative temporal distances between interactions.

[Q4] Constant Stream: *The constant stream has no time scale, the optimal partition is the whole timeline.*

A constant stream is the stream for which the same set of edges occurs at every time step. Let DG be a dynamic graph over the constant stream. Let DG' be a coarsening of DG. Then, the quality function of the coarsening is better:

$$\forall DG, DG', Q(DG) < Q(DG').$$

Consequently, the optimal DG^* with respect to the quality function Q on the constant stream is the aggregation of the whole timeline.

[Q5] Stream with no Temporal Scale: *The quality function is the same for any partition of the stream with no temporal scale.*

Formally, let DG and DG' be any two dynamic networks corresponding to any two partitions $\mathcal{P}$ and $\mathcal{P}'$ of the stream with no temporal scale. Then, the quality function of the two dynamic networks should not be different:

$$\forall DG, DG', |Q(DG) - Q(DG')| \geq \epsilon.$$

We have given only one explicit example of a stream with no temporal scale: the $DynUR$ stream, and the statement in this axiom should be considered in this context.

[Q6] Temporal Shift Invariance: *A shift of the time line of a temporal stream, does not drastically change the quality of the dynamic network. The optimal partition of the stream is independent of the time line's starting point.*

Formally, let $[0, \ldots, T]$ be the time line of the temporal stream E_t. Let $[\Delta, \ldots, T + \Delta]$ be the new timeline shifted by parameter $\Delta > 0$. Let DG^* represent the optimal dynamic network for E_t and $DG^{*'}$ be the optimal dynamic network for the shifted temporal stream. Then, the change of the quality function due to the shift is small:

$$|Q(DG^{*'}) - Q(DG^*)| < \epsilon.$$

An interval in a partition of the stream identifies a temporal cohesive unit in the dynamic network, similar to the notion of a building block. It is more important that interactions happen a specific time apart rather than where in the stream they happen. Temporal shift invariance is a strong assumption, considering the absolute time dependency of complex systems. For example, empirical analysis of mobile phone calls (34) shows that, especially for a finer time scale, the start time of the partition matters a lot. In general, it is of great interest to characterize interaction streams (and their generative mechanisms) that are independent of the start time, and those streams that are highly sensitive.

5.2 Discussion

Axioms [Q1] and [Q2] view the underlying dynamic process and the temporal structure it contains as a sequential process. These axioms specify when the ordering of the interactions matters and when it is just temporal noise. Axioms [Q3] through [Q5] are formalizations of intuitive observations about the TSI problem. As a collection, the axioms presented here are not exhaustive of all the properties characterizing the temporal scale of interaction streams. Yet, these axioms provide a starting point for formally addressing the TSI problem in a rigorous and consistent manner. The axioms serve as external evaluators of the quality of a dynamic network, in the sense that they do not depend on the type of the partitioning algorithm used, objective function, or the generative process of the temporal stream. Therefore, the usefulness of the axiomatic framework is two-fold. The framework can be used to:

1. evaluate the performance of a partitioning algorithm in a unbiased way.

2. provide a taxonomy of different partitioning algorithms.

In Chapter 6, we present two such algorithms and analyze their behavior in the context of the axiomatic framework.

CHAPTER 6

INHERENT TIMESCALE

This chapter presents two approaches for identifying the "right" window of aggregation/"optimal" partition of the temporal stream. We use insights and formulations developed in Chapters 4 and 5 to design two heuristics that apply our current understanding of notions of noise, structure and quality of a dynamic network as they relate to the problem of temporal scale.

6.1 Information Theoretic Approach

In this section, we relate the notion of the "right" temporal scale to that of information embedded in a dynamic network. Not surprisingly, the notion of information in networks is ambiguous and ill-defined, especially when we take into consideration the complex and non-linear structures that networks can represent(48; 8; 29; 40). The complimentary notion, noise (lack of structure) in temporal streams suffers from the same ambiguity of definition. In Chapter 4, we looked at specific examples of noise and defined an instantiation of this concept. The results from this analysis gave us insights on how unstructured temporal interactions are different from structured ones. We are now ready to apply some of these insights to design a constructive approach for identifying the temporal scale which is most informative.

The concept of "information" embedded in interaction streams is general and includes stationarity as one of its special cases. Here we take the perspective of information theory, where we define the notions of "meaningful" or "informative" in the context of the best trade-off between noise and compressibility inherent in a dynamic network. This approach offers the benefit of defining the "right" temporal scale irrespective of a specific learning objective. The resulting dynamic network generated from this analysis should, therefore, be a good representation for the underlying stream of interactions in a wide variety of applications. One way of evaluating results of this approach would be to use a network where we know the right timescale. It is at that scale that events such as communities or anomalies can be detected. An orthogonal method of evaluation would be to consider the performance of the approach in the context of the axiomatic approach. In the following section, we will use some of the ideas discussed here and present an information theoretic-based heuristic for the TSI problem.

6.1.1 TWIN Heuristic

The TWIN (**T**emporal **W**indow **I**n **N**etworks) heuristic uses graph-theoretic measures as proxies of different aspects of network structure. Given a temporal stream of edges and a graph-theoretic measure (Figure 11), the heuristic generates time series of graphs (dynamic graphs) at different levels of aggregation. It then computes the variance and compression ratio for each time series. Finally, the algorithm analyzes the compression ratio and variance as functions of window size and selects the window size for which compression ratio and variance are close or equal to each other. TWIN analyzes a variety of graph-

theoretic measures, definitions of which can be found in Chapter 3. The list of measures is not exhaustive of all measures that can be used to analyze network structural properties. Rather, the goal is to illustrate the effect of aggregation window on the behavior of a wide range of measures each revealing unique and interesting properties of the network. Analysis of this framework can easily be extended to other network measures not mentioned here. In addition to the analysis of different network measures, TWIN considers different types of temporal streams, from synthetic streams discussed in detail in Chapter 4 to more complex real-world streams described in Chapter 3.

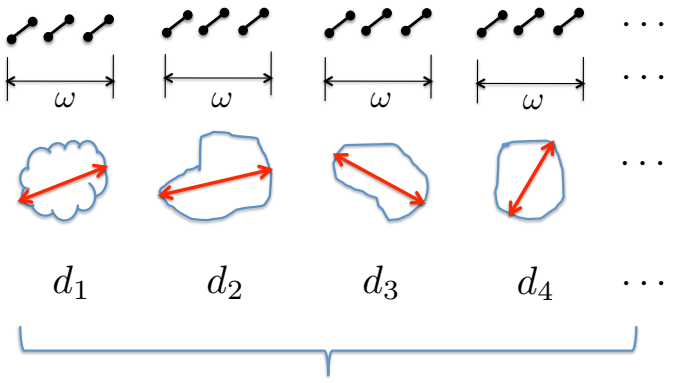


Figure 11: Illustration of computation of variance and compression ratio for a given level of aggregation (ω), and a given graph-theoretic measure d (diameter).

We now formally define the measures that can be used to quantify noise and information in a dynamic network. Given a fixed window of aggregation $\underline{\omega}$, let $\underline{DG_\omega}$ be the corresponding

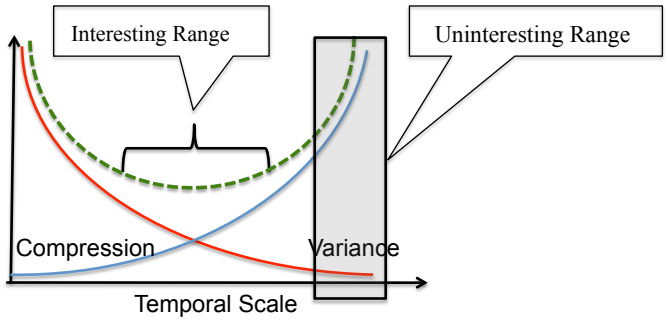


Figure 12: Trade-off plot of noise and compression measures.

dynamic network. Let $\underline{f}$ be a graph-theoretic measure such as density, clustering coefficient, etc., and $\underline{\mathcal{F}_\omega}$ its corresponding time series computed over DG_ω, the dynamic network defined over the uniform partition of the time line:

$$\mathcal{F}_\omega(DG_\omega) = [f(G_1), f(G_2), ..., f(G_t), ..., f(G_{\frac{T}{\omega}-1})]$$

Let $\underline{V(\mathcal{F}_\omega)}$ be the variance of $\mathcal{F}_\omega$:

$$V(\mathcal{F}_\omega) = \frac{1}{\frac{T}{\omega-1}} \sum_{t=1}^{\frac{T}{\omega-1}} [f_\omega(G_t) - \mu(f_\omega)]^2,$$

where $\mu(\mathcal{F}_\omega) = \frac{1}{T} \sum_{t=1}^{\frac{T}{\omega-1}} f_\omega(G_t)$.

We can think of the $V(\mathcal{F}_\omega)$ as a measure of noise present in $\mathcal{F}_\omega$. Large values of variance indicate $\mathcal{F}_\omega$ changes drastically in time making it hard to distinguish between the occurrence

of a meaningful change and a noise effect. On the other hand, small values of variance indicate $\mathcal{F}_\omega$ is smooth and noise is removed.

Compression Ratio deserves a little bit more careful consideration to measure computationally. In general, let u be the length of the string representation of $\mathcal{F}_\omega$ in some representation system. Let c be the length of the compressed representation of $\mathcal{F}_\omega$ in the same representation system produced by a data compression algorithm. Let $\underline{R(\mathcal{F}_\omega)}$ be the compression ratio of $\mathcal{F}_\omega$ defined as:

$$R(\mathcal{F}_\omega) = \frac{u}{c}$$

$R(\mathcal{F}_\omega)$ is one of the ways to represents information encoded in $\mathcal{F}_\omega$. A small value of $R(\mathcal{F}_\omega)$ represents a lot of randomness or noise in signal $\mathcal{F}_\omega$, while a large value of $R(\mathcal{F}_\omega)$ comes as a result of redundancies in $\mathcal{F}_\omega$. In the information theoretic sense, redundancies correspond to low entropy and low entropy corresponds to high information.

Variance and compression ratio are not mathematical compliments of each other, but they do have opposite behavior as functions of window size. As illustrated in Figure 12, when we increase the value of ω, we expect the variance to decrease and compression rate to increase. There is a region in Figure 12 where variance is very small and compression is very high. However, low variance and high compression in this region are achieved artificially by aggregating the underlying stream at too coarse a scale, so that all the critical temporal information is removed. We call the range of window sizes that fall in this region *"uninteresting"*. Instead, there is a range of window sizes for which we can expect both

relatively low variance and relatively high compression levels of time series F_ω. This insight allows us to formulate the process of finding the range of appropriate discretization window sizes as a search problem guided by the values of variance and compression.

Algorithm 1 TWIN Heuristic:

Require: Temporal stream E_t, graph-theoretic measure f, user-defined "goodness measure" γ, maximum window size analyzed ω_{max}.
Ensure: List of appropriate window sizes $\{\omega\}$
1: **for** $\omega = 1$ to ω_{max} **do**
2: Compute the time series of graphs $DG_\omega : [G_1, G_2, ..., G_t, ..., G_{\frac{T}{\omega}-1}]$
3: Compute the time series $\mathcal{F}_\omega : [f(G_1), f(G_2), ..., f(G_t), ..., f(G_{\frac{T}{\omega}-1}]$
4: **if** $V(\mathcal{F}_\omega) - R(\mathcal{F}_\omega) < \gamma$ **then**
5: Output ω
6: **end if**
7: **end for**

6.1.2 Experimental Setup

In this section, we first demonstrate the importance of the window of aggregation in the analysis of dynamic networks. We then evaluate the results of our heuristic across datasets coming from different domains (simulations, sociology, communication) in order to show the breadth of the applicability of the results. For these datasets, the ground truth comes either from domain knowledge or embedded structure in the case of synthetic data. Through simulation and ground truth that comes from domain knowledge, we show that our heuristic produces meaningful and consistent levels of temporal aggregation. We compare our method

with GraphScope, in the context of event detection, and FFT analysis for identifying the scale of temporal dynamics and demonstrate that our results are equally robust and some times better when detecting events and patterns in dynamic networks. Finally, we illustrate that TWIN's performance is consistent with respect to axioms $Q1, Q2, Q3$ (Section 5).

For the synthetic datasets, we generate both realizations of the $DynUR$ stream and the $DynMix$ stream. We generate the $DynUR$ stream using number of vertices $n = 100$, probability of an edge $p = 0.01$ and number of time steps $T = 100$. We generate the $DynMix$ stream using two alternating versions of the beta distribution. For the first distribution, we use shape parameters $\alpha = 5, \beta = 1$ which lead to the generation of a dense temporal stream. For the second distribution, we use parameters $\alpha = 1, \beta = 3$ which lead to the generation of a sparse temporal stream. We alternate each distribution every 20 steps. Once the probability of an edge is generated, it is kept the same through out the 20 time steps. The intention here is to obtain as persistent a stream as possible by minimizing changes due to the randomness of the generative process.

6.1.3 Experimental Results

6.1.3.1 Real-world Dynamic Networks

We begin with Enron dataset and radius as a measure. Figure 13 shows the plot of variance V and compression ratio R of network radius times series as a function of time. Note that R increases as ω increases, while V (overall) decreases. The plot suggests that an appropriate window for analyzing the radius of the Enron network is in the range of 4-7 days, where variance is relatively small and compression is relatively high. Figures 14

displays the time series of radius for the Enron dataset at $\omega = 1$ (high aggregation level), $\omega = 5$ (aggregation level within the range 4-7 days suggested by TWIN) and $\omega = 12$ (coarse aggregation level), correspondingly. As seen in Figure 14(a), the drastic variations of the radius time series at 1-day aggregation make it impossible to detect any pattern in the dynamics of email exchanges of the Enron employees. As we increase the aggregation window to 5 days (Figure 14(b)), some peaks corresponding to important events in the lifetime of the Enron company become clear. For example, the peak at timestep 950 (Event 1) represents the time when Karl Rove sold off his energy stocks, the peak at timestep 1100 (Event 2) represents the unsuccessful attempt of Dynegy to acquire the bankrupt Enron, while the peak at timestep 1150 (Event 3) represents the resignation of Enron's CEO in January 2002, and the beginning of the FBI investigation. When we aggregate the dynamic network beyond the 4-7 day range, as in Figure 14(c), we notice that the time series becomes smoother, but at the same time, some critical temporal events are lost. For example, the collapse of the Dynegy deal represented by a sharp peak at aggregation level $\omega = 5$ is not identifiable anymore. Similar behaviors were observed for measures computed on the Reality Mining and Haggle datasets (illustrated on Figure 15(a), and Figure 15(b)). Also, summaries of the appropriate window ranges for the Enron, Reality Mining, Haggle, Grevy's and Plains datasets are given in Table I.

6.1.3.2 Synthetic Dynamic Networks

Figure 16 displays results of TWIN for the $DynUR$ stream. Note that in the trade-off plot (Subfigure (a)) variance does not show a trend of decreasing as $t \to T$. Therefore,

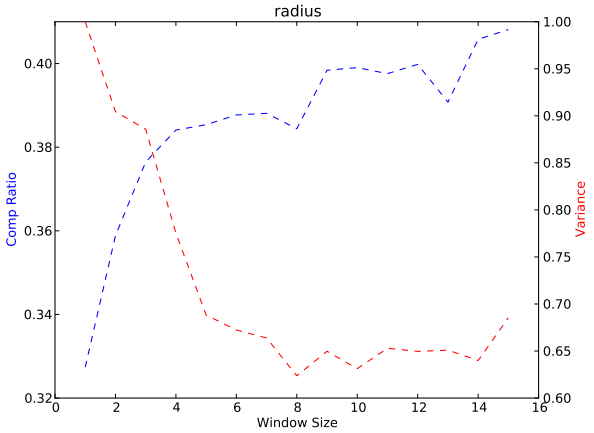
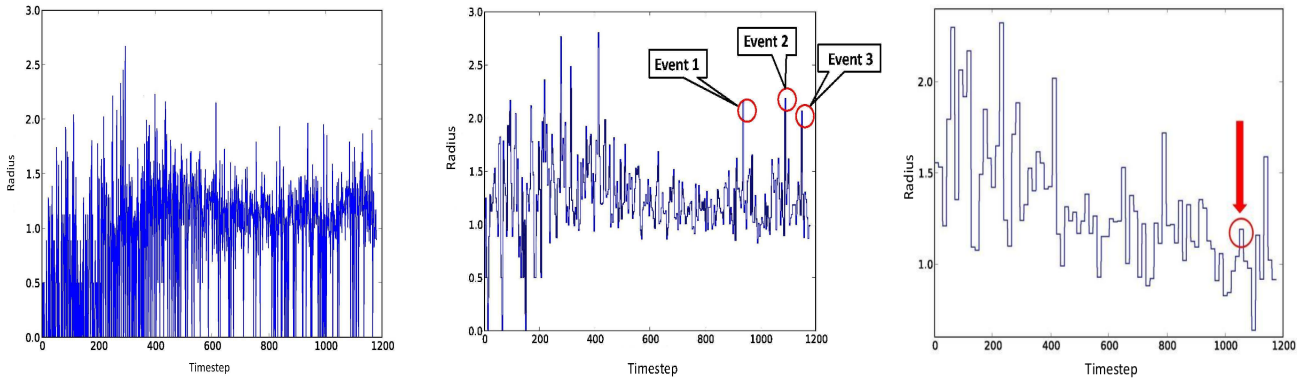


Figure 13: TWIN's trade-off plot of variance (V) and compression rate (R) of network radius with respect to window of aggregation ω.

TWIN is not able to identify an optimal window of aggregation. In Figure 16 (b) we see the time series of the graph density function for $DynUR$ at different windows of aggregation ($\omega = 1, 2, 3$). Just as we expect, the time series look similarly noisy across the different scales. Figure 17 shows the results of TWIN for the $DynMix$ dataset. We notice that at window $\omega = 20$, we get a sharp decrease of the variance followed by the stabilization of its value. $\omega = 20$ corresponds to half the period of the $DynMix$ stream (when the alternation of the two probability functions happens). Furthermore, note that both scales $\omega = 5$ and $\omega = 20$ capture the periodicity of the stream correctly and have smoothed out the noise present at $\omega = 1$. Yet, $\omega = 20$ seems qualitatively better, because at this scale the peaks of the time series are just as pronounced as in $\omega = 1$. This is exactly the result that would



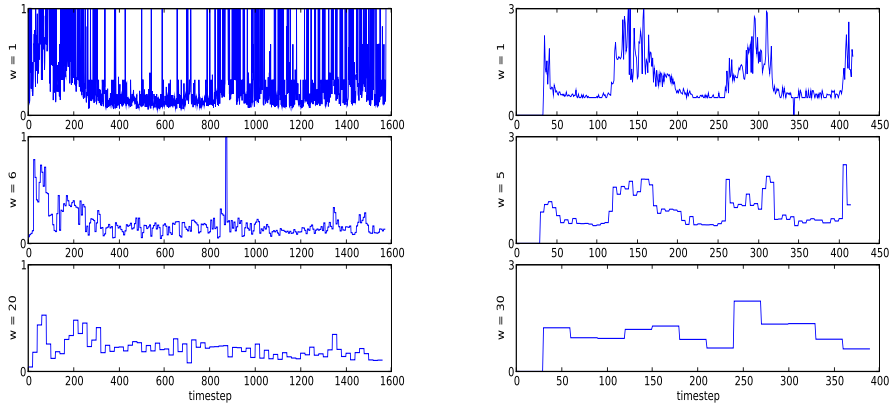
(a) $\omega = 1$ day (b) $\omega = 5$ days (c) $\omega = 12$ days

Figure 14: Network radius time series for the Enron dataset at three levels of aggregation: (a) fine level of aggregation, $\omega = 1$ day, (c) coarse level of aggregation, $\omega = 12$ days, (b) the right level of aggregation, $\omega = 5$ days.

like to achieve when we aggregate at the right scale. We would like to smooth out only the noise, without affecting the quality of the actual signal (information) in the data.

6.1.3.3 Comparison with Graphscope and FFT Method.

Graphscope analysis on the Enron dataset partitions the time line on intervals that vary from 2 weeks to 6 weeks, during the eventful period of November 2001-May 2002. Some of the major events are captured using this partitions. There are however, several important events that get smoothed out and can not be spotted when analyzing the time series aggregated at such coarse levels (Figure 18 (a)). Since GraphScope focuses on variations of graph compression levels, it is the magnitude of change in the graph structure that drives the time line partitioning. TWIN analyzes the regularity of compression levels

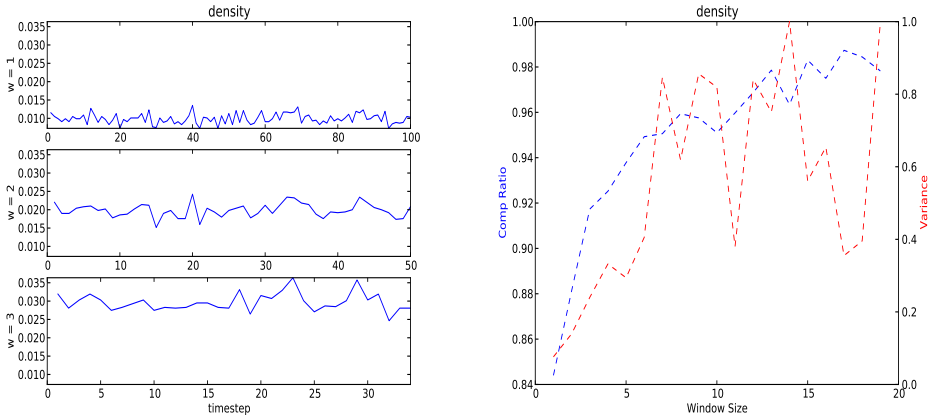


(a) (b)

Figure 15: Network density for the Reality Mining dataset (a) and Haggle dataset (b) at three levels of aggregation: too fine level of aggregation(top picture), the right level of aggregation (middle picture) and too coarse level of aggregation (bottom picture).

of different metrics on the graph, and therefore, it is the rate of change, not the magnitude, that will have the most effect in the aggregation.

A nice feature of the Graphscope heuristic is the fact that it generates a non-uniform partitioning of the time line. The non-uniform partitioning is a more realistic representation of real-world interaction streams which are commonly characterized by bursty behavior (3; 32). On the other hand, Graphscope determines this partitioning for a fixed aggregation step and it does not take into account the effect the aggregation step has on the computation of the compression cost. The estimation of persistent structures leading to the low compression costs is highly sensitive to the size of aggregation level. TWIN overcomes this dependency



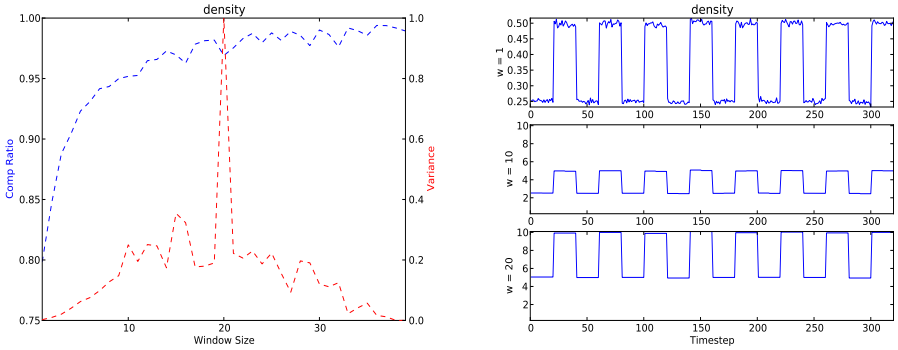
(a) Trade-off plot for the variance and compression ratio of network density.

(b) Time series of graph density at different windows of aggregation.

Figure 16: TWIN's results for the measure of network density of the $DynUR$ stream.

by analyzing the persistent nature of the stream across different scale, and picking those scales where persistence is more pronounced.

Figure 18 (b) shows the clique number for the Enron dataset when the underlying temporal graph is aggregated at 4 days, as recommended by the TWIN heuristic, and 7 days, following the predominant cycle identified by the FFT analysis. We notice that at 7 days important peaks of the signal are not as easy to identify or completely disappear. Since the rate of change in a temporal graph does not follow a simple pattern, using periodicity (i.e. FFT method) to determine the right aggregation levels might not always be appropriate.



(a) Trade-off plot for the variance and compression ratio of graph density.

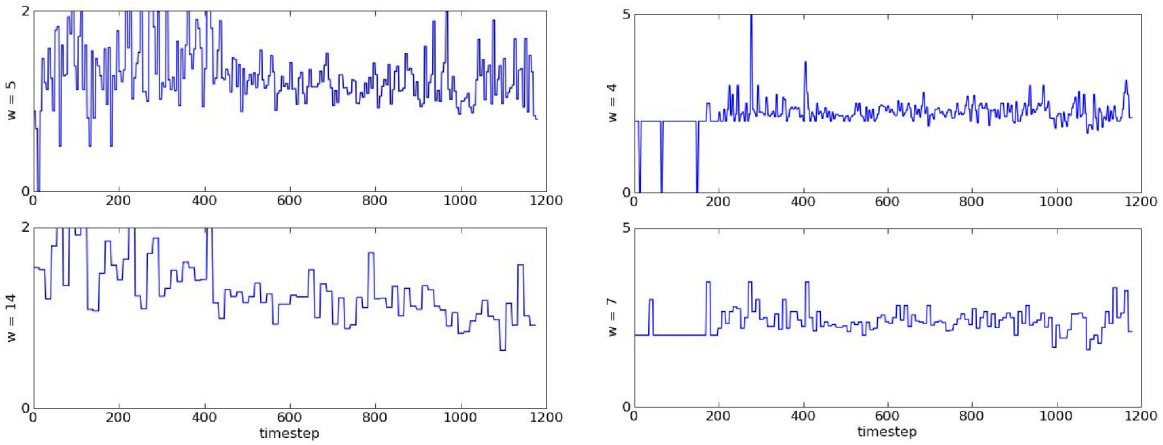
(b) Time series of graph density at different windows of aggregation.

Figure 17: TWIN's results for the measure of network density of the *DynMix* stream.

6.1.4 TWIN in the Context of the Axiomatic Framework

6.1.4.1 Perturbation Analysis

Recall that in Chapter 5 we described several desired properties of the optimal partition and optimal dynamic network for a given temporal stream. In particular, axioms [Q1] and [Q2] describe when the re-ordering of interactions in time matter and when it is an artifact of the data collection process. [Q1] states that at the optimal partition, any re-ordering within an interval of this partition does not change the quality of the dynamic network. On the other hand, [Q2] states that re-orderings across intervals of the partition can significantly change the quality of the dynamic network. Definition 5.1.1 formalizes the two types of the permutation function: π_w (permutations within an interval) and π_a (permutations across intervals).



(a) Clique number for Enron dataset at $\omega = 4$ days and $\omega = 7$ days

(b) Average path length for Enron dataset at $\omega = 5$ days and $\omega = 14$ days

Figure 18: Comparison of TWIN heuristic to Graphscope heuristic (a) and FFT method (b).

We will now analyze the performance of TWIN in the context of axioms [Q1] and [Q2]. Let ω^* be the output of TWIN for the given temporal stream E_t. Let $E_t^w = \pi_w(E_t, \mathcal{P}_\omega^*)$ be the perturbed stream using function π_w, where a fraction η of edges has been re-ordered in time. Let $E_t^a = \pi_a(E_t, \mathcal{P}_\omega^*)$ be the perturbed stream using function π_a, where a fraction θ of edges has been re-ordered in time. If ω^* is indeed an optimal window of aggregation, applying the TWIN heuristic to stream E_t^a should theoretically produce the same optimal window of aggregation ω^*. In contrast, applying the TWIN heuristic to stream E_t^w will significantly change the optimal window of aggregation ω^*. Given the noise present in real-world networks and the fact TWIN is a heuristic rather than an optimization algorithm, the perturbations due to π_w are not guaranteed to produce identical answers, but hopefully the

TABLE I: Results of TWIN heuristics for the Enron, Reality Mining, Haggle.

Measure	Enron Dataset	Reality Mining	Haggle
Density	4-5	5-10	-
Number of connected components	3-5	10-14	35-50
Size of the giant component	-	45-55	38-58
Diameter	6-8	12-25	3-45
Radius	4-7	15-35	10-45
Average Path Length	6-10	20-30	3-20
Clustering Coefficient	-	12-25	38-55
Clique Number	2-4	-	-
Spectral Gap	-	-	5-10
Graph Compression Ratio	-	2-5	5-35

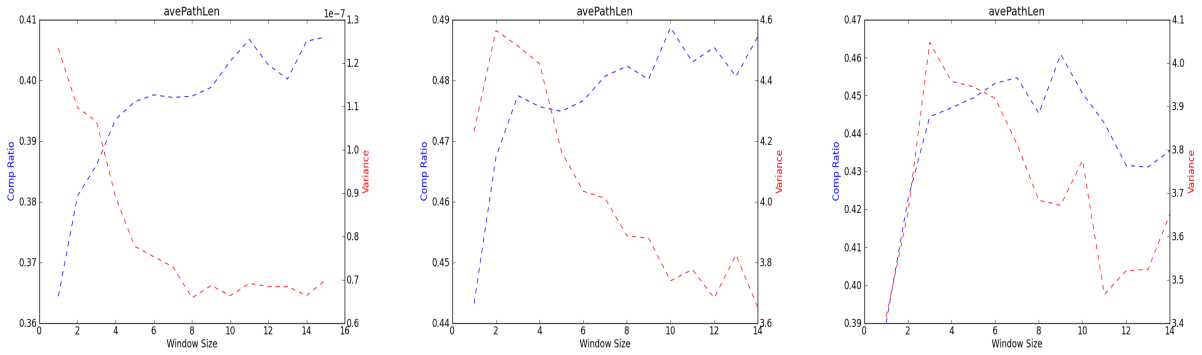
answers will be close enough. On the other hand, perturbations due to π_a should produce a noticeably different answer. We also expect the amount of perturbation to have an affect on the how much TWIN-s output changes. Naturally, very high levels of perturbation can change the topological structure of the underlying network so much that the output of TWIN might change drastically to reflect the new topological structure. In Figure 19, we give an illustration of TWIN's behavior under the two different types of edge permutations for the measure of graph average path length for the Enron dataset. The perturbation factors used for the analysis are $\eta = \theta = 10\%$. In Figure 19(a), we notice that TWIN identifies window size $\omega = 4$ as the right level for aggregation. When the input is the stream with edge temporal re-ordering within intervals of length 4 (Figure 19(b)), TWIN identifies the same window of aggregation as in the original stream. On the other hand, when TWIN analyzes

the stream with re-orderings across intervals of length 4 (Figure 19(c)), it selects $\omega = 6$ as the right window of aggregation. Furthermore, from a qualitative point of view, we notice that the variance curve does not nicely stabilize anymore as $t \to T$. Figure 20 demonstrates similar results when perturbation analysis is applied to the Haggle and Reality Mining datasets.

Figure 21 shows how TWIN behaves when re-orderings of edges are applied to the $DynUR$ and $DynMix$ streams. Note that in the case of the $DynUR$ stream, re-orderings within and across intervals lead to almost identical outputs from TWIN (Figure 21 (b), (c)). This behavior is just what we should expect, since the $DynUR$ stream has no temporal scale (i.e. the temporal order of edges is not important). In the case of $DynMix$, we do observe the consistency of the output when re-orderings of the edges happen within the same interval (Figure 21 (d) (e), (f)). However, we do not observe the expected sensitivity to re-orderings across intervals. This behavior is unvarying for all the other graph theoretic measures (e.g. density, average degree, radius) and for other values of perturbation parameters ($\eta = \theta = 15\%, 20\%$).

6.1.4.2 Oversampling Analysis

We now analyze the behavior of TWIN in the context of the oversampling axiom [Q3]. Recall that the oversampling axioms states that oversampling the initial temporal stream at a uniform rate should not change the quality of the resulting dynamic network in a substantial way. In order to test the performance of TWIN with respect to this axiom, we oversample a given stream and compare the results on the new and original stream. We



(a) Original (b) Within Perturbations (c) Across Perturbations

Figure 19: Average Path Length of original network (a), network perturbed within each partition (b), and network perturbed across partitions (c) for the Enron dataset.

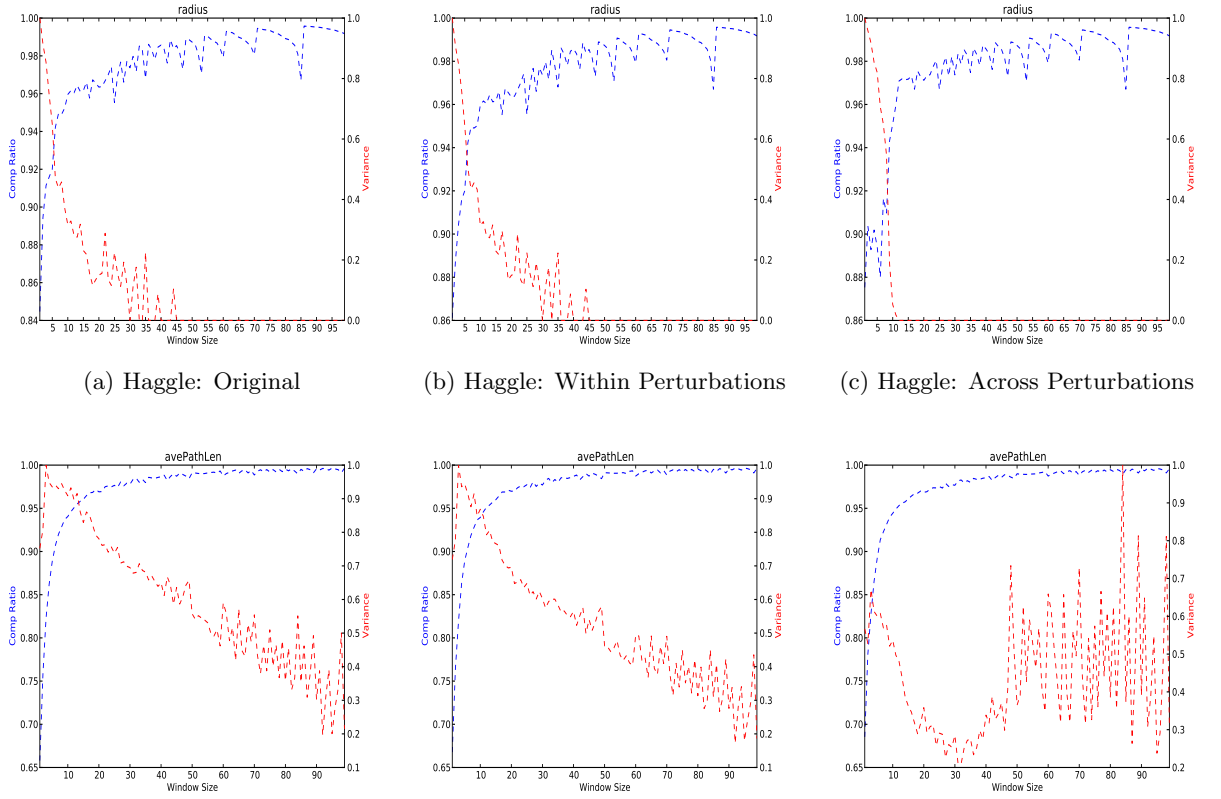
would expect that TWIN selects the same (or very similar) window of aggregation in both cases.

Figure 22 shows the behavior of TWIN when the *DynMix* stream is oversampled by a factor of $\alpha = 4$. Note that the trade-off plot for the average path length measure stretches correspondently by a factor of 4. Similar results can be observed for the Reality Mining stream (Figure 23).

6.1.5 Summary

The empirical framework of TWIN offers the following contributions:

- It gives a quantitative trade-off criterion for identifying the appropriate window size for discretizing a dynamic networks. By choosing windows of aggregation that balance between the minimization of noise and loss of temporal structural information,



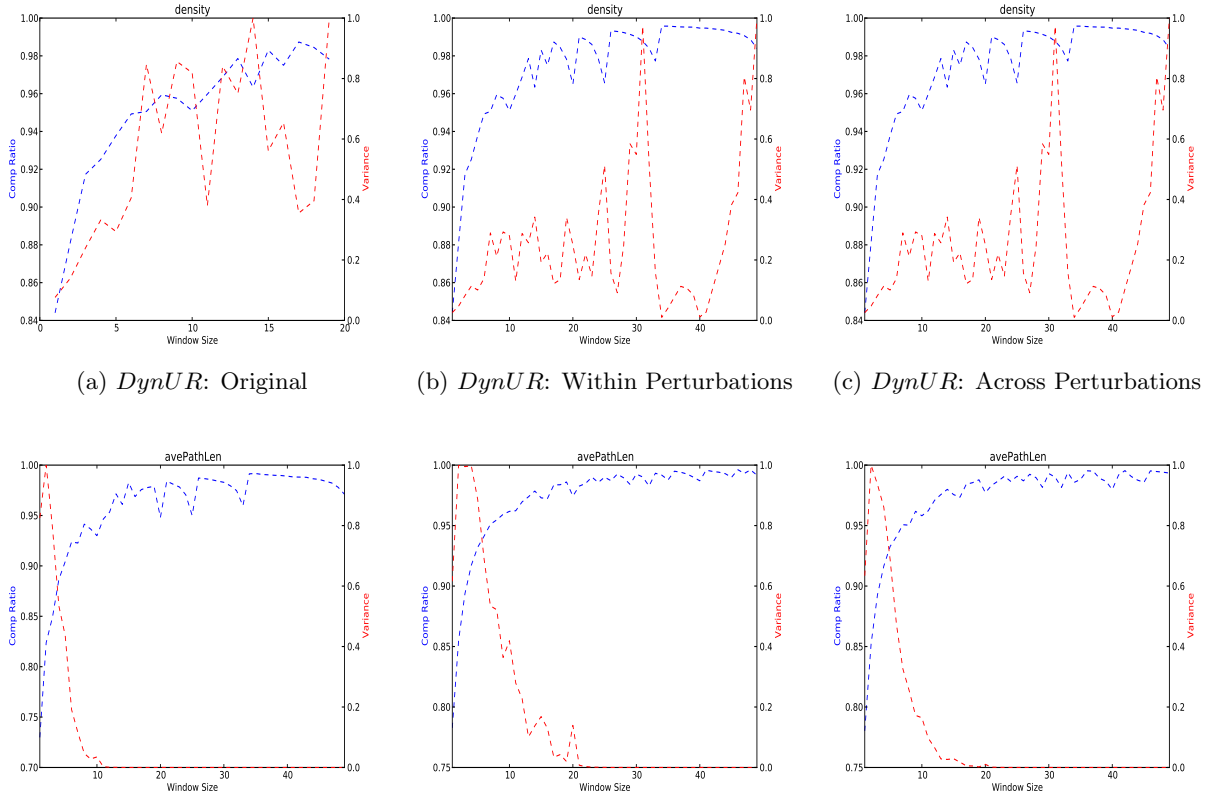
(a) Haggle: Original (b) Haggle: Within Perturbations (c) Haggle: Across Perturbations

(d) Reality Mining: Original (e) Reality Mining: Within Perturbations (f) Reality Mining: Across Perturbations

Figure 20: Perturbation analysis of TWIN's performance for the Reality Mining and Haggle datasets.

TWIN's approach offers a systematic framework to empirically discover interesting network dynamics that would otherwise be lost.

- The framework presented here does not restrict the analysis to one network statistic. We show that different aggregation levels are appropriate for different network measures. Not only this is not a drawback of our method, but it is a desirable fea-



(a) $DynUR$: Original (b) $DynUR$: Within Perturbations (c) $DynUR$: Across Perturbations

(d) $DynMix$: Original (e) $DynMix$: Within Perturbations (f) $DynMix$: Across Perturbations

Figure 21: Perturbation analysis of TWIN's performance for the $DynUR$ and the $DynMix$ datasets.

ture, since each measure reveals distinct properties of the network. Furthermore, it is another illustration of the fact that interesting network behavior happens at various temporal resolutions and our method automatically reveals those interesting temporal scales.

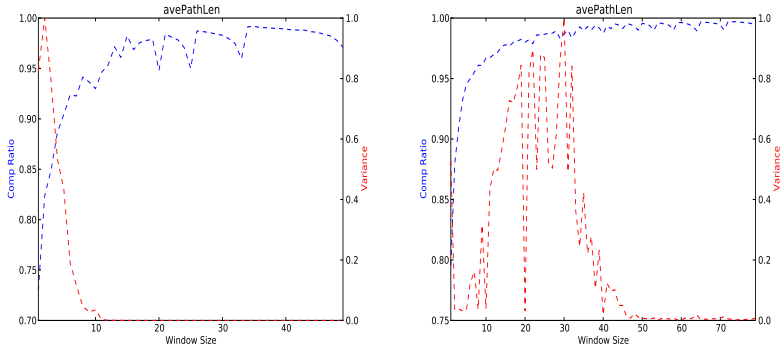


Figure 22: Average path length of original $DynMix$ stream(a), and oversampled stream(b) with oversampling factor $\alpha = 4$.

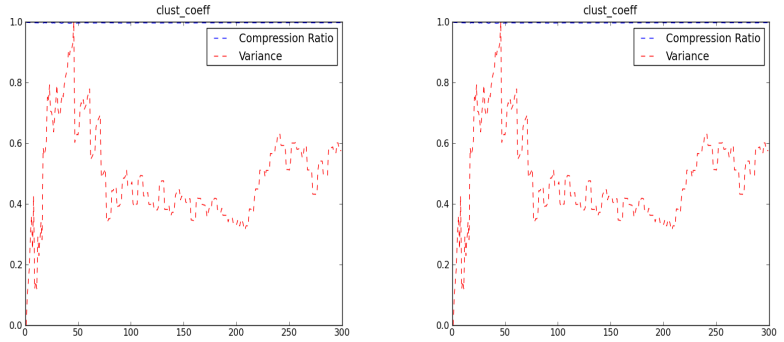


Figure 23: Clustering Coefficient of original Reality Mining stream (a), and oversampled stream (b) with oversampling factor $\alpha = 5$.

- The heuristic produces consistent results for datasets arising from different domains and different underlying network dynamics.
- Finally, TWIN performs well when analyzed in the context of the axiomatic approach. In particular, TWIN's performance is stable with respect to temporal re-ordering within its optimal window of aggregation and sensitive to temporal re-ordering across.

This indicates that TWIN is able to select windows of aggregation that correspond to inherent scales of the dynamics of the underlying system. In addition, TWIN's performance is not sensitive to uniformly increasing the oversampling factor of the interaction stream. This demonstrates TWIN's robustness with respect to noise resulting due to oversampling of the stream.

6.2 Persistence-based Approach

Interactions observed fleetingly along the time line of a stream are often not interesting and they usually indicate that the data collection process is noisy. On the other hand, interactions that persist for awhile truly represent what is more "essential" for the underlying system. What is, then, the "right" temporal scale that can capture the persistence of structure in time, while smoothing out the random fluctuations? Consider, for example, the event of the onset of the FBI investigation of the Enron Corporation and the flurry of emails that followed. Intuitively, we would like to analyze this collection of emails together as one temporal unit. This temporal unit represents the right granularity to capture the temporal causality in this scenario.

In social network analysis, one point of view considers persistent interactions over time as defining more complex sociological structures such as relationships or kinship (46). In another context, the notion of persistence is critical in extending the static definition of communities to that of dynamic communities (64). The common thread of the definition of persistence across the different disciplines and applications within them is that persistence is a property that allows us to construct a network with the "core" interactions, discarding the

noisy transient interactions. Therefore, the notion of persistence lends itself to yet another formulation of the "right" temporal scale: the temporal scale at which the underlying network is most persistent in time.

6.2.1 DAPPER Heuristic

In general, persistence can be thought as a local property of temporal data since temporal dependencies tend to weaken over longer time intervals. Here we describe a new approach for the TSI problem that uses the notion of persistence as a local qualitative measure. The DAPPER (**D**ynamic **A**pproach for identifying **P**attern **Per**sistence) heuristic exploits the local persistence (quality) of interaction streams to navigate towards a globally optimal partition of the timeline. This approach can generate optimal partitions that are not uniform along the time line. In this respect, the DAPPER heuristic is a departure from the TWIN heuristic and offers a more realistic representation of temporal streams.

6.2.1.1 Measuring Local Persistence

There are many ways to measure local temporal persistence. One way is to use the changes in edge frequency values as a proxy. By focusing on the edge frequencies over time, we avoid having to make any assumptions about other more complex structures in the underlying network. Without any additional information or specified objective, using the information of volume of edges in time is the best thing we can do. At a greater computational cost, the frequency approach can be extended to any fixed substructure that is deemed more relevant for the analysis. We intuitively expect that the network structure that persists over time is a manifestation of more or less the same set of edges occurring

consistently. Therefore, such persistent structure is characterized by small changes of the edge frequency vector, but equally important, these small changes persist over time. Once we have the individual edge occurrence counts, we express the total frequency score of the underlying network as a linear combination of individual edge frequencies. This assumes independence of edge occurrences. Assuming edge independence is a starting point and can be generalized later. Intuitively, high persistence of network structure implies persistence of edge frequency values, but the converse is not necessarily true. In Figure 24, we illustrate such a situation. Intervals p_i and p_i' both have the same frequency values, yet interval p_i' has higher persistence. In Section 6.2.5.2, we discuss two intuitive properties (internal consistency and local monotonicity) that could potentially be used as part of a post-hoc analysis to filter out any intervals that might have high frequency values, but are not persistent enough.

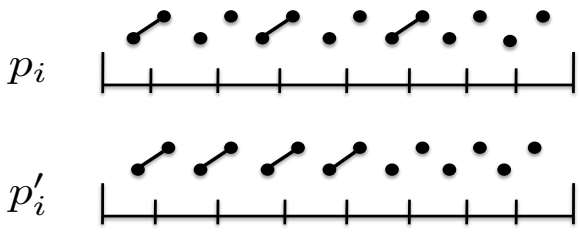


Figure 24: Two examples of intervals with same frequency, but different persistence pattern.

Let ω represent the length of the interval we will use to obtain an initial uniform partition $\mathcal{P}_\omega$ of the time line. Let $p = [l, l+\omega)$ be an interval in $\mathcal{P}_\omega$ and $freq(p)$ the frequency vector representing the number of times each edge $e \in E$ occurs in p:

$$freq(p) = (\frac{1}{\omega} \sum_{l \leq t < l+\omega} X_{ijt})_{|E|},$$

where X_{ijt} is the indicator variable representing edge e_{ij} being present at time t. Note that we normalize the number of edge occurrences within an interval by the length of the interval. The normalization step is important when we compare the quality of intervals of different lengths. Also, the normalized edge counts directly relate to the notion of edge probabilities and, in this sense, we can think of function $freq$ as an estimation of the probability function generating the stream.

Let fd be the frequency difference function representing the amount of change in edge frequencies between two consecutive time intervals:

$$fd(p_{i-1}, p_i) = ||freq(p_{i-1}) - freq(p_i)||_{\mathbf{p}}$$

Figure 25 gives an illustration of how DAPPER computes the endpoints of two consecutive intervals along the time line. Note that unlike the implementation of the TWIN heuristic, here consecutive intervals are overlapping and parameter s controls the amount of overlap. Overlapping consecutive intervals are a natural generalization that is intended to give smoother measurements of the temporal characteristics of the stream. In the current

DAPPER implementation we compute the l_1 norm of the fd vector ($\mathbf{p} = 1$). Other norms can be used, but careful consideration needs to be given to the effect the value of $\mathbf{p}$ has on representing the magnitude of change in edge frequencies.

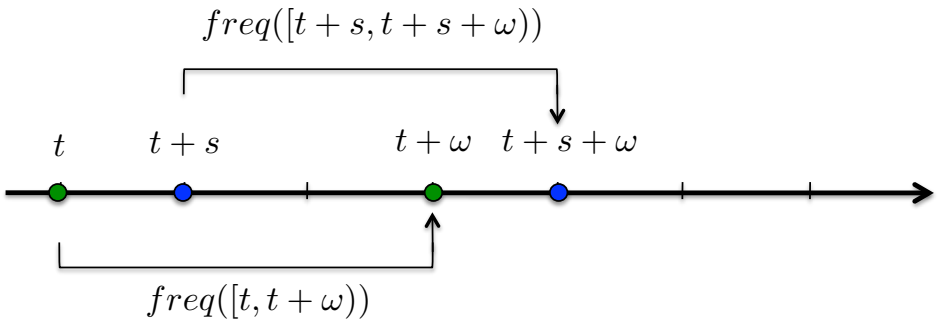


Figure 25: Illustration of the temporal bounds of the edge frequency function $freq$ for window size $\omega = 3$ and shift parameter $s = 1$.

Let LM be the set of *local* maxima:

$$LM = \{i : fd(i) >= fd(i+/-j), i-r \leq j \leq i+r\}$$

where r is the local parameter. The definition of *local* depends indirectly on the value of ω. For $\omega = 1$, *local* means comparing the value of fd with r timesteps before and r timesteps after. As the value of ω increases, so does the temporal scope of what is considered *local*. The choice of the local parameter, similarly to the shift parameter s is related to the trade-off

between computational efficiency and accuracy. In the current implementation of DAPPER we are using $r = 1$.

Following, we characterize two types of intervals whose quality we want to capture:

Type 1: $(l, r) \cap LM = \emptyset$. There are no local maxima inside interval (l, r). Type 1 intervals partition $[0, \ldots, T]$, and $LM \cup \{0, T\}$ equals the set of endpoints of the Type 1 intervals. If we modify function fd by adding a little noise, then LM may gain some values in (l, r), but $\min(fd)$ on (l, r) will be quite similar to the old value of $\min(fd)$. This motivates the following definition for the quality function q_1 of Type 1 intervals:

$$q_1 = \frac{\min\{fd(l), fd(r)\} - \min\{fd(x) : l < x < r\}}{r - l}.$$

In other words, we make a rectangle with left side $x = l$, right side $x = r$, top $y = \min\{fd(l), fd(r)\}$, and bottom $y = \min\{fd(x) : l < x < r\}$, and q_1 is the slope of its diagonal.

Type 2: $(l, r) \cap LM \neq \emptyset$. There are local maxima inside interval (l, r). Let m be the value in (l, r) such that $fd(m)$ is maximized. Note that $m \in LM$. (For simplicity, we assume that m is unique.) We then make a rectangle with left side at $x = l$, right side at $x = r$, bottom at $y = fd(m)$ and top at $y = \min\{fd(l), fd(r)\}$. Intuitively,

when this box is deeper, we have a better interval. To quantify this, we define quality function q_2 of Type 2 intervals as follows:

$$q_2 := \frac{\min\{fd(l), fd(r)\} - fd(m)}{r - l}.$$

We hope that Type 1 and Type 2 intervals can capture all the locally high-quality intervals along the time line. However, we can expect that not all Type 1 and Type 2 intervals will necessarily be of high quality. The presence of noise in the stream may cause Type 1 intervals to be too small (too fine). If there are long high-quality intervals, Type 1 will not suffice to capture them. Furthermore, if the persistent structure of the stream is multi-scalar, Type 1 intervals can only possibly detect persistence at the smallest scale. Our hope is that Type 2 intervals will address some of these challenges.

Some additional work is required to synthesize the highest quality intervals among Type 1 and Type 2 intervals that can give us the global partition of the time line (explained in more detail in Section 6.2.2). Note that the requirement that we output a partition might mean that not all the intervals in the final answer will be of the highest quality. If instead of one global partition, we were interested in multi-scalar output, a different kind of analysis is required. One idea would be to generate a set of "fuzzy" partitions, where at each temporal scale, only high-quality intervals are reported. If these intervals do not cover the whole time line, we label the remaining

regions of the time line as "fuzzy". This is a direction that we would like to further explore in the future.

6.2.2 Outline of the DAPPER Heuristic

In this section we give an outline of the steps the DAPPER heuristic follows to compute Type 1 and Type 2 intervals, and how it then uses their quality measures to generate a global partition of the time line.

1. **Generate potential breakpoints using the concept of local maxima:**

   (a) **Compute Type 1 Intervals:** Consider the frequency difference function $fd(t)$ as t advances from 0 to T to find the set of local maxima LM (as well as Type 1 intervals) and the minimum value of $fd(t)$ for each Type 1 interval. Last, since we have the $min(fd(t))$, we can also compute the value of q_1 for each Type 1 interval.

   (b) **Compute Type 2 Intervals:** Let LM^* be a copy of LM. We will manipulate this list to find and create the Type 2 intervals. First, we sort LM so that $fd(t)$ is in non-decreasing order. At each iteration, we consider $m \in LM$, starting with m where $fd(m)$ is smallest. At each step, we remove m from LM^*. Let l be the element in LM^* preceding m and let r be the element following m. We output the Type 2 interval $[l, r]$ associated with m and its quality value q_2.

2. **Synchronize Type 1 and Type 2 intervals to generate a partition:**

   (a) Take the union of Type 1 and Type 2 intervals and their corresponding q values.

(b) Sort the intervals by their q-values in non-increasing order, with ties broken arbitrarily.

(c) Initialize the set of breakpoints $B := \emptyset$.

(d) Iterate: Starting with the interval with the highest quality value (either q_1 or q_2), add the endpoints of the corresponding interval. Let $[l, r]$ be the next unprocessed interval. If the endpoints of the unprocessed interval fall inside any of the intervals already added to B, ignore the interval and move to the next unprocessed interval.

(e) When the procedure quits: if $B = \{b_1, \ldots, b_k\}$ with $b_1 < \ldots < b_k$, then our final answer is the set of intervals $[0, b_1), [b_1, b_2), \ldots, [b_k, T]$.

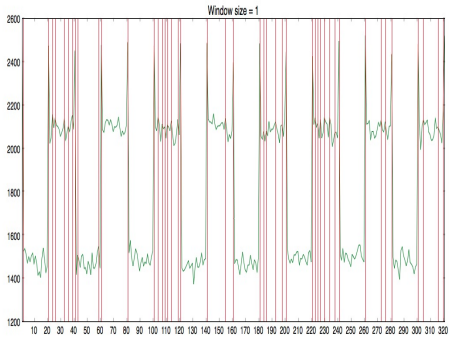
6.2.3 Experimental Setup

We generate instances of the $DynMix$ Stream using two alternating probability distributions: the beta distribution and gaussian distribution. We alternate each distribution every 20 steps. Once the probability of an edge is generated, it is kept the same through out the 20 time steps. The intention here is to obtain as persistent a network as possible by minimizing changes due to the randomness of the generative process. For the beta distribution, we use shape parameters $\alpha = 1, \beta = 3$ which lead to the generation of a sparse temporal stream. The gaussian distribution, on the other hand, generates a $DynUR$-like temporal stream.

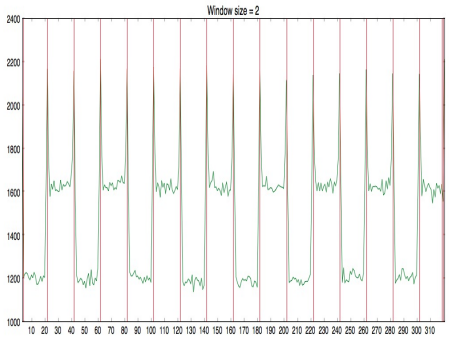
6.2.4 Experimental Results

Figure 26 shows the results of the DAPPER heuristic for the $DynMix$ Stream for three windows of aggregation $\omega = 1, 2, 3$. At $\omega = 1$, DAPPER correctly identifies the critical breakpoints at every 20 time steps corresponding to the times the underlying probabilistic process alternated from the beta to the gaussian distributions. However, at this temporal scale, DAPPER generates additional breakpoints in temporal regions where there are no changes to the underlying generative process. Interestingly, these regions, all correspond to high frequency differences associated with the $DynUR$ stream. Since the $DynUR$ stream is essentially noise, it overwhelms the analysis of DAPPER at $\omega = 1$. When we increase the window size to $\omega = 2$, DAPPER is able to identify the correct partitioning. We are able to see the effect aggregation has in smoothing out some of the noisy fluctuations in edge frequency values. DAPPER performs equally well for window sizes $\omega > 2$.

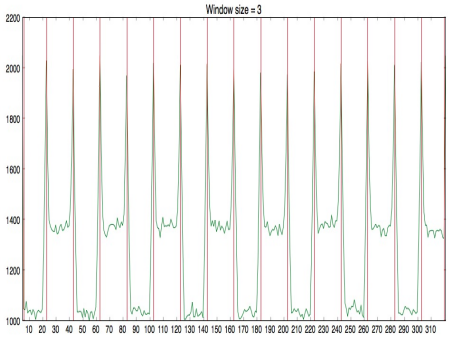
Figure 27 shows the results of the DAPPER heuristic for the Haggle Stream. Recall from Section 4.3.7, the Haggle dataset represents proximity-based interactions of participants in the IEEE Infocom '06 conference for a period of four days. Note that for window size $\omega =$ (which corresponds to a 10 minute interval), DAPPER generates a very fine partition of the time line. The over-partitioning during the four pronounced peaks (corresponding to the four conference days) is somewhat to be expected due to the multi-scalar nature of the Haggle dataset. One can think of 20 minute talks, 30 minute talks, or morning and afternoon sessions as equally "natural" temporal intervals for the partitioning of the time line. However, for the temporal intervals corresponding to the evening and night times



(a) $\omega = 1$



(b) $\omega = 2$

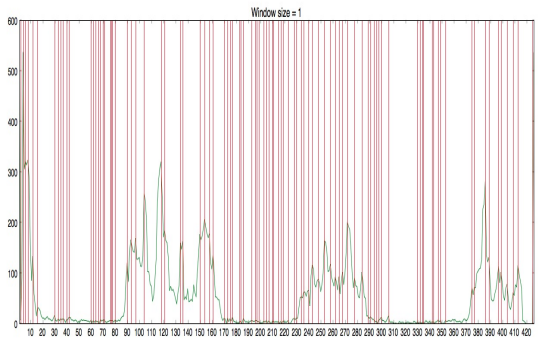


(c) $\omega = 3$

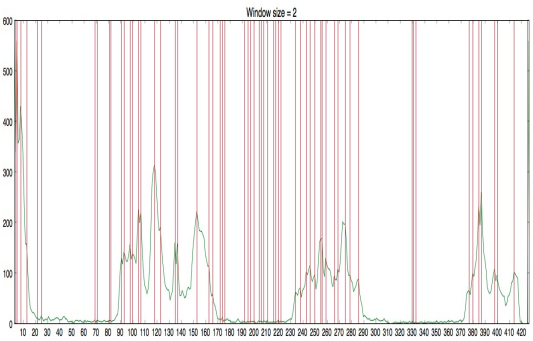
Figure 26: Partitioning of the *DynMix* stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line.

(regions with very low and stable frequency difference values), DAPPER still over-partitions at $\omega = 1$. This problem seems to be corrected as the value of ω is increased and we notice that for $\omega = 4$ (40 minute intervals), we see a clear separation between the day and night frequency patterns (Figure 27(c)). Also, note that some of the finer partitions at this scale, do indeed correspond to intervals of length 20 minutes, 30 minutes, 50 minutes and about 3-4 hours. The results map consistently to the temporal organization of the conference and

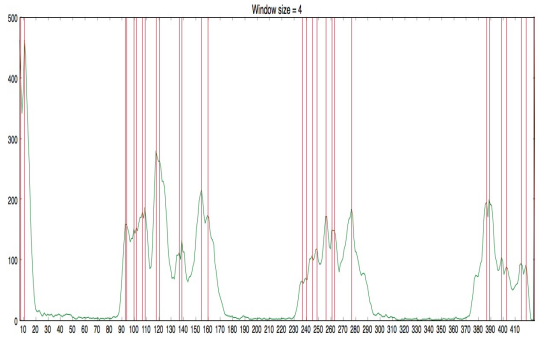
in this sense, DAPPER captures the right scale of the underlying dynamics of the Haggle dataset.



(a) $\omega = 1$



(b) $\omega = 2$



(c) $\omega = 4$

Figure 27: Partitioning of the Haggle stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line.

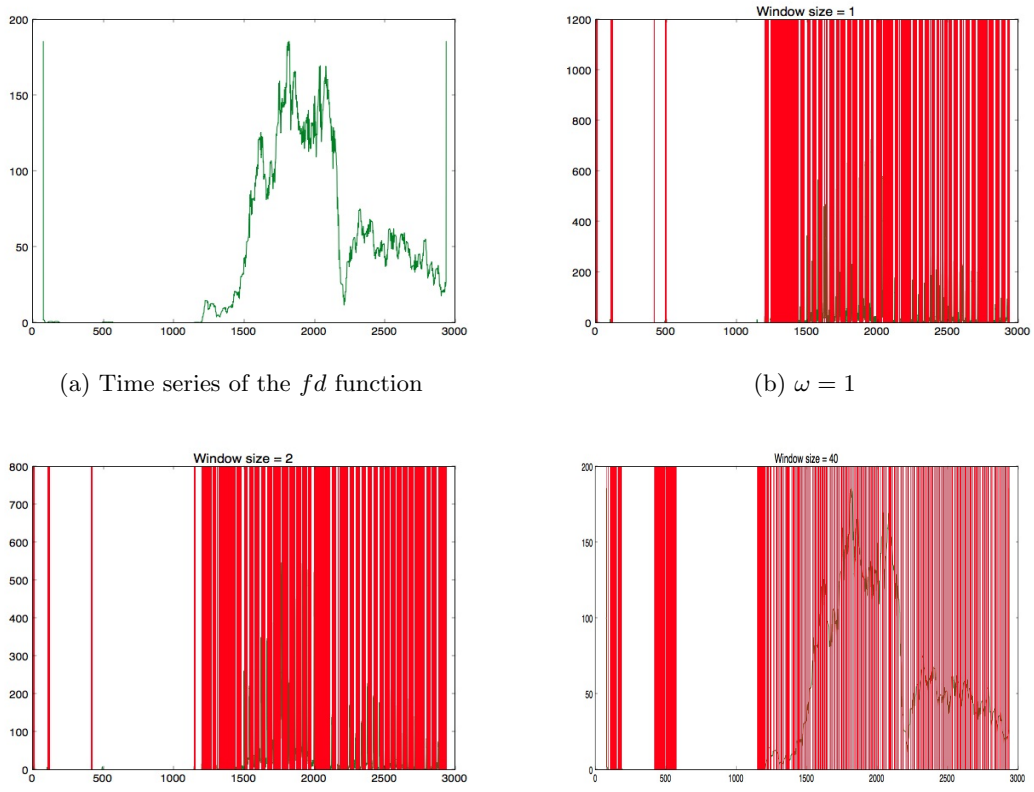
Figure 28 shows the results of DAPPER for the Reality Mining Stream. Recall from Section 4.3.7, the Reality Mining stream represents proximity-based interactions of students at MIT during an academic year. The original temporal scale of the temporal stream is 4 hours. As illustrated by Figure 28, DAPPER generates a very fine partition of the time line. It seems the increase of ω is not able to smooth out the temporal noise present in this dataset. Even for a relatively high window size ($\omega = 40$ corresponding to about 1 week along the Reality Mining time line), the partition is very similar to what we get for $\omega = 1$.

6.2.5 Discussion

6.2.5.1 Challenges for Larger Values of ω

As discussed in Section 6.2.1.1, when the interaction stream has a lot of noise, the frequency difference function fd will often be fairly high for very small values of ω (in particular for $\omega = 1$). If there are two consecutive intervals with different behavior, but which are not very different from each other, or where the change from one interval to the other is gradual, then this will not show up very strongly in fd, even in the absence of noise. Thus, in the presence of noise, the break between the two intervals may be impossible to detect (with $\omega = 1$).

By considering a larger ω, the noise is averaged out (just as when we are dealing with oversampling), but the difference in the consecutive intervals is not averaged out, so it becomes easier to detect true breakpoints along the time line. We gave an illustration of this observation in Section 6.2.4, when we applied DAPPER on the $DynMix$ and Haggle datasets. However, there are some complications that arise when considering larger values of



(a) Time series of the fd function

(b) $\omega = 1$

(c) $\omega = 2$

(d) $\omega = 40$

Figure 28: Partitioning of the Reality Mining stream by the DAPPER heuristic. The red vertical lines represent the partitioning points along the time line.

ω. Currently, we consider overlapping consecutive intervals ($s = 1$), but such an approach is computationally expensive. In the future, we would like to compare non-overlapping consecutive intervals of the form $[k\omega, k\omega+\omega)$ and $[k\omega+\omega, k\omega+2\omega)$. This form of partitioning creates a few issues:

1. We might miss a good breakpoint. For example, for $\omega = 10$, we might find a breakpoint at $t = 80$, but it turns out that the real breakpoint is at $t = 79$ or $t = 83$. One possible way to fix this problem would be to search locally (to the left and right of the breakpoint) to see if there is any nearby points that are better choices for the breakpoint.

2. Some values of ω may be too big to detect any real structure. If we do not recognize that ω is too big, we might assume that the output indicates a good breakpoint, when in fact it is just caused by looking for data at the wrong scale. A useful feature of the algorithm would be to automatically decide whether a particular value of ω is too big and should be ignored (or should be ignored for another reason). Note that this means that one possible output for each ω should be "Nothing useful found" (For example, in the case of the $DynUR$ stream, this should be the case for every window size.)

3. Synthesizing results across different scales to get a global partition of the time line is not straightforward. This issue is connected to the previous two issues. This statement presupposes that for each ω, we will get some output that is independent of what the algorithm does for other ω, and the outputs are only combined in the end. The outputs could be a set of intervals, each one with a quality Q, or it could even be a partition of the entire time line. Perhaps a better approach would be to integrate the process of partitioning at different scales rather than trying to merge the outputs at the end.

6.2.5.2 Desired Properties of a Local Quality Measure

In Section 6.2.1.1 we defined two ways to measure the local quality of intervals with respect to the notion of temporal persistence (q_1, q_2). Similarly to the definition of the global quality Q of a partition (discussed in Section 5), there are many ways to going about defining what local quality means in the context of temporal scale of interaction streams. Here we discuss two intuitive properties that we would like any local quality function to have. More formally, let q represent a persistence-based local quality function. Then q takes as input an interval p_i in the timeline $[0, \ldots, T]$, the set of edges E_t^i that occur during this interval, and outputs a quality score about the persistence of edges that occur during p_i:

$$q : (p_i, E_t^i) \rightarrow \mathbb{R}^+$$

The first property we discuss here summarizes the observation that persistent structure is spread out in a sequential fashion (see Figure 24 for illustration). The second property states our preference of choosing longer persistent intervals.

[p1] *Internal Consistency:* Let p^* be an interval in an optimal (with respect to q) partition $\mathcal{P}^*$ of temporal stream E_t. Consider a "big enough" subinterval $p_i \subseteq p^*$, such that $|p_i| > |p^*|/2$ (Figure 29(a)). Then, with high probability, the quality of subinterval p_i is close to the quality of the interval p^*:

$$\forall p_i \subseteq p^* \text{ s.t. } |p_i| > |p^*|/2, \quad \Pr[|q(p^*, E_t^*) - q(p_i, E_t^i)| \leq \epsilon] \geq 1 - \delta.$$

[p2] *Local Monotonicity:* Let p^* be an interval in an optimal (with respect to q) partition $\mathcal{P}^*$ of temporal stream E_t. Consider two "big enough" subintervals $p_i, p_j \subseteq p^*$, such that $|p_i|, |p_j| > |p^*|/2$ and $|p_i| \geq |p_j|$ (Figure 29(b)). Then, with high probability, the bigger subinterval has higher quality:

$$\forall p_i, p_j \subseteq p^* \text{ s.t. } |p_i|, |p_j| > |p^*|/2, |p_i| \geq |p_j|, \quad \Pr[q(p_i, E_t^i) \geq q(p_j, E_t^j)] \geq 1 - \delta$$

We could think of q_1 and q_2 as estimations of the rate of change of the edge probability function. Intuitively, we would expect that during an interval with"optimal persistence", the parameters of the edge probability functions stay the same throughout the interval, and therefore its rate of change is essentially constant. In this sense, we hope both q_1 and q_2 can capture well at least property **p1**.

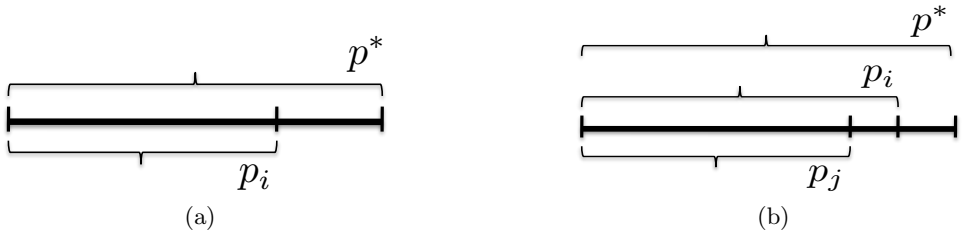


Figure 29: Illustration of the internal consistency and local monotonicity properties.

6.2.5.3 Comparison of the DAPPER Heuristic with TWIN and Graphscope Heuristics

The use of a local quality function represents an important difference between the DAPPER approach and the TWIN approach. The TWIN heuristic selects the best partition based on a global criterion of quality. Both variance and compression rate are qualitative measures of the whole timeline. In contrast, the DAPPER heuristic takes the view that a global partition with high quality is a sequence of intervals with high quality. In addition, the TWIN heuristic assumes a uniform partition of the timeline. The underlying implication is that data occurs and is collected at a uniform rate along the time line. Such an assumption might be too strong when considering real-world systems. For example, we would expect the email communications of Enron employees to be more dense during the week days, and much more sparse over the weekend. Naturally, a non-uniform partition of the time line would be a better representation of scenarios like this. The DAPPER heuristic does not restrict the optimal solutions to the set of uniform partitions of the time line. To this extent, it represents a more general and more realistic mode for partitioning temporal streams.

The DAPPER heuristic is similar to the Graphscope heuristic (61) in that it uses the notion of the persistence to segment the time line. Graphscope identifies change points based on identifying intervals with "good" compression level. The compression function used by this heuristic is related to the persistence-based quality function that the DAPPER heuristic uses.

There is, however, an important difference between the approach of DAPPER and Graphscope. Graphscope computes the "best" partition of the timeline on one swipe of the temporal stream (i.e. one fixed temporal scale). On the other hand, the DAPPER heuristic looks at persistence across the timeline, as well as, persistence across temporal scale. In this sense, DAPPER can be thought as a generalization of the Graphscope approach. The justification for the DAPPER approach is two fold: **1.** the initial temporal resolution of the data might not be the most appropriate scale at which persistence is revealed, **2.** persistence structure usually persists at different scales. Therefore, merging persistent intervals across scales is more robust; it will ignore those intervals that are persistent only in few scales (persistence might be an artifact, rather than an inherent characteristic), and it will merge intervals that "consistently" appear as persistent.

CHAPTER 7

FUTURE WORK AND CONCLUSIONS

7.1 Directions for Future Work

The TSI problem for interaction streams has only until recently received the deserved attention. The analytical framework we have presented in this thesis focused on preparing the groundwork for rigorously defining and solving the TSI problem. We would like to extend this framework in a several directions:

7.1.1 Analysis of Special Cases of Interaction Streams

As illustrated in this thesis, the analysis of special cases of interaction streams, where we either understand the structural or generative properties of the temporal stream, or we know what the "right" temporal stream should be (e.g. domain knowledge), can lead to important insights for the TSI problem. We would like to continue in this direction by studying more general and complex cases of interaction streams:

Markov-based streams: These are streams where the probabilities of edges occurring within a window of aggregation only depend on the probabilities of edges occurring in the previous window of aggregation. This kind of generative model explicitly incorporates temporal dependencies often observed in real-world interaction streams. The Markov model is well-studied in many contexts and it has the potential to allow us to say something precise for the TSI problem.

Streams with topological interaction dependence: In the current framework, we assume interactions in the network occur independently of each other. This is a strong assumption, especially when we try to apply our analysis to real-world networks. As a first step towards improving our framework, we can start with introducing a simple case of edge dependence, such as dependence between pairs of interactions. The goal is then to study implications of this generalization on the TSI framework.

Streams with fractal-like temporal structure: The topological structure of fractal-like networks (e.g. Kronecker graphs) has recently gained a lot of interest (39). Similarly, we would like to study and define the temporal scale properties of fractal-like interaction streams. Intuitively we would expect such streams to be invariant with respect to the aggregation process (i.e. no matter the level of aggregation, the result is always a time series of self-similar graphs). To this extend, the $DynUR$ stream can be thought as a special case (although not a very interesting one). More generally, we would like to study streams that have the fractal behavior with respect to temporal scale, yet they have topological structure embedded in them.

7.1.2 Extensions to Perturbation Analysis Framework

In this thesis, we have developed two techniques for testing the quality of the solution from a TSI algorithm: temporal re-orderings and oversampling. There are multiple directions to extend and generalize these techniques:

Topologically non-uniform temporal re-orderings: In the current framework, we select the subset of edges to be re-ordered uniformly at random. An extension to this

technique would be to bias the selection by taking into account the topology of the interactions (i.e homophily, communities).

Modification of the rate of occurrence through oversampling The rate of oversampling in the current framework is uniform along the time line. A possible generalization would be to consider non-uniform over-sampling.

7.1.3 Objective-based Formulations of TSI Problem

We have illustrated how dynamic networks have inherent rhythms that govern their dynamics. This is one natural way to define what is interesting about them. An alternative way would be to define the interestingness of the network based on what is useful about them. This leads to an orthogonal approach that is application driven. For example, the identification of the most frequent sub-graphs, or the identification of dynamic communities are useful applications that give us meaningful insights about the network.

It is often the case that the algorithms designed for these applications rely on the fact that the temporal dynamics of the network are represented at the appropriate temporal scale for their analysis. For example, defining dynamic communities depends on identifying co-occurring sets of interactions that persist together in time (64). Naturally, the concept of what is considered persistent depends on the temporal scale at which the network is analyzed. Since different patterns can develop at different temporal scales, it is natural to ask the following question: At what temporal scale does a pattern of interest become detectable in the dynamic network? What is the right dynamic network representation that

captures what is essential about the pattern of interest? We briefly give some preliminary intuition of the TSI problem in a objective-specific setting.

7.1.3.1 Algorithmic-specific Formulation of TSI Problem

Given a specific learning objective on a dynamic network, and an algorithm designed to achieve it, the goal is to identify the temporal scale at which the performance of the algorithm is maximized. More formally:

Let $\mathcal{A}$ be a learning algorithm taking as input DG and giving as output a solution $\mathcal{O}$. Let $\mathcal{O}^*$ be the optimal solution for the learning objective. Then we can define the TSI problem with respect to algorithm $\mathcal{A}$ as follows:

$$\langle \mathcal{P}^*, DG^* \rangle = \underset{\mathcal{P}, DG}{\operatorname{argmin}} \left[\mathcal{A}(DG) - \mathcal{O}^* \right]$$

7.1.3.2 TSI Problem for Dynamic Community Identification

The notion of a dynamic network community is very elusive. There are a lot of definitions offered in the existing literature. Intuitively, a community is a cohesive collection of nodes. The notion of cohesiveness if often defined as interactions among nodes inside the collection having greater strength or frequency than interactions with nodes outside the collection (65). This intuitive notion can be thought as a governing rule and a discerning characteristic of what is a network community.

The introduction of the temporal component in the analysis of communities allows us to observe the community as it changes and involves in time. Nevertheless, the property

of cohesiveness stays intact in time. Intuitively, a dynamic community is easier to detect when temporal perturbations that are noise do not change too much the structure of the community. At the optimal temporal scale, temporal re-orderings of edges within the same network partition (temporal unit) do not cause drastic changes in the dynamic network structure. However, temporal re-orderings of edges across partitions fundamentally modify the underlying structure. It is of interest to investigate if the notion of "stability under temporal perturbation" could be a useful criteria in identifying the optimal temporal scale for community identification.

7.1.3.3 TSI Problem for Dynamic Link Prediction

The Dynamic Link Prediction problem can be summarized by the following two objectives: 1) Given an already seen interaction, predict when it will occur again; 2) given a historic stream of interactions, predict whether an unseen interaction will occur (illustrated in Figure 30). Prediction is a learning task that relies on identifying a generalizable representation of "historic" data. In the context of TSI problem we ask: at what temporal scale is the "historic" dynamic network best for predicting future interactions?

Since the temporal delay between interactions within a partition is artifactual, aggregation at the right window allows for separation of noise from the essential delay. As consequence of this property, the problem of link prediction based on time delays (37) becomes identical to the problem of link prediction based on the actual time occurrence of interactions (41). In this context, the optimal partition of the interaction stream can be de-

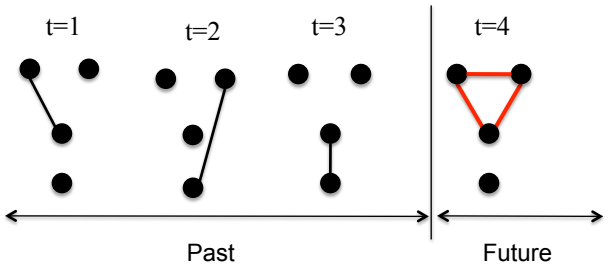


Figure 30: Illustration of Dynamic Link Prediction Problem

fined as the partition at which the discrepancies between the two link prediction approaches are minimized.

7.2 Conclusions

There is both an intuitive understanding and mounting empirical evidence that the temporal scale of interaction streams plays an important role in their analysis. Moreover, it is clear that many interaction streams have a set of relevant scales and that those may change over time. Some of the scales are inherent to the dynamics of the interactions, while others are only relevant depending on the context of the analysis performed on the stream. All of this makes the problem of identifying and inferring the temporal scale of interaction streams important, yet equally elusive and difficult to state.

In this thesis, we brought together the various interpretations of the concept of temporal scale and pointed out the evidence that supports those interpretations. We formalized the problem of the temporal scale inference, defined some intuitive properties of the "right" temporal scale, and proposed two heuristics for solving the problem.

We hope this thesis brings the problem of temporal scale of interaction streams to the forefront of research consciousness, makes the problem explicit, and provides the tools for making progress in this area. Understanding the rhythm of interacting systems is not only necessary for the proper analysis of these systems, but will provide us with the fundamental insight into what makes these systems tick. It is an important, challenging, and worthy endeavor.

CITED LITERATURE

1. Ackerman, M., Ben-David, S., and Loker, D.: Towards property-based classification of clustering paradigms. In NIPS, pages 10–18, 2010.

2. Baldock, K., Memmott, J., Ruiz-Guajardo, J., Roze, D., and Stone, G. S.: Daily temporal structure in african savanna flower visitation networks and consequences for network sampling. Ecology, 92, March 2011.

3. Barabasi, A.-L.: Bursts: The Hidden Pattern Behind Everything We Do. Dutton Adult, 2010.

4. Barabási, A.-L. and Albert, R.: Emergence of scaling in random networks. Science, 286(5439):509–512, Oct 1999.

5. Barron, A. R., Rissanen, J., and Yu, B.: The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory, 44(6):2743–, 1998.

6. Ben-David, S. and Ackerman, M.: Measures of clustering quality: A working set of axioms for clustering. In NIPS, pages 121–128, 2008.

7. Bender-deMoll, S. and McFarland, D. A.: The art and science of dynamic network visualization. Journal of Social Structure, 7(2), 2006.

8. Berger-Wolf, T. Y. and Saia, J.: A framework for analysis of dynamic social networks. In Proc. 12th ACM SIGKDD on Knowledge discovery and data mining, pages 523–528, New York, NY, 2006. ACM.

9. Blonder, B., Wey, T. W., Dornhaus, A., James, R., and Sih, A.: Temporal dynamics and network analysis. Methods in Ecology and Evolution, Blackwell Publishing Ltd.

10. Butts, C. T.: An axiomatic approach to network complexity. In The Journal of Mathematical Sociology, volume 24, pages 273–301, 2000.

11. Caceres, R. S., Berger-Wolf, T., and Grossman, R.: Temporal scale of processes in dynamic networks. In IEEE 11th ICDM Workshops, pages 925–932, 12 2011.

12. Chapanond, A., Krishnamoorthy, M., and Yener, B.: Graph theoretic and spectral analysis of enron email data. Computational & Mathematical Organization Theory, 11:265–281, Oct 2005.

13. Chung, F., Lu, L., and Vu, V.: Spectra of random graphs with given expected degrees. Proceedings of the National Academy of Sciences, 100(11):6313–6318, 2003.

14. Chung, F.: Spectral Graph Theory. CBMS. AMS, 1997.

15. Clauset, A. and Eagle, N.: Persistence and periodicity in a dynamic proximity network, 09 2007.

16. Clauset, A., Shalizi, C. R., and Newman, M. E. J.: Power-law distributions in empirical data. SIAM Review, 51(4):661–703, 2009.

17. Cross, P. C., Lloyd-Smith, J. O., and Getz, W. M.: Disentangling association patterns in fission-fusion societies using african buffalo as an example. Animal Behaviour, 69, 2005.

18. Eagle, N. and Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing, V10(4):255–268, May 2006.

19. Erdos, P. and Renyi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci, 5, 1960.

20. Feldmann, A., Gilbert, A. C., Willinger, W., and Kurtz, T.: The changing nature of network traffic: Scaling phenomena. Computer Communication Review, 28:5–29, 1998.

21. Fischhoff, I. R., Sundaresan, S. R., Cordingley, J., Larkin, H. M., Sellier, M.-J., and Rubenstein, D. I.: Social relationships and reproductive state influence leadership roles in movements of plains zebra, equus burchellii. Animal Behaviour, 73(5):825–831, May 2007.

22. Gao, Q., Li, M., B, M., and Vitányi, P.: Applying mdl to learning best model granularity, 2000.

23. Hinde, R. A.: Interactions, relationships, and social structure. Man, 11:1–17, 1976.

24. Holme, P.: Network reachability of real-world contact sequences. Phys. Rev. E, 71:046119, Oct 2004.

25. Holme, P. and Saramäki, J.: Temporal networks. ArXiv e-prints, 08 2011.

26. Hu, B., Rakthanmanon, T., Hao, Y., Evans, S., Lonardi, S., and Keogh, E. J.: Discovering the intrinsic cardinality and dimensionality of time series using mdl. In ICDM, pages 1086–1091, 2011.

27. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabasi, A. L.: The large-scale organization of metabolic networks. Nature, 407(6804):651–654, Oct 2000.

28. Karsai, M., Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Barabási, A.-L., and Saramäki, J.: Small but slow world: How network topology and burstiness slow down spreading. Phys. Rev. E, 83:025102, Feb 2011.

29. Kempe, D., Kleinberg, J., and Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci., 64(4):820–842, 2002.

30. Keogh, E. J., Chu, S., Hart, D., and Pazzani, M. J.: An online algorithm for segmenting time series. In ICDM, pages 289–296, 2001.

31. Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Saramäki, J., and Karsai, M.: Multiscale analysis of spreading in a large communication network. Journal of Statistical Mechanics: Theory and Experiment, 3:5, 03 2012.

32. Kleinberg, J.: Bursty and hierarchical structure in streams. In 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 07 2002.

33. Kleinberg, J.: An impossibility theorem for clustering. In Advances in Neural Information Processing Systems, pages 446–453. MIT Press, 2002.

34. Krings, G., Karsai, M., Bernharsson, S., Blondel, V. D., and Saramäki, J.: Effects of time window size and placement on the structure of aggregated networks. ArXiv e-prints, 02 2012.

35. Kumar, R., Novak, J., and Tomkins, A.: Structure and evolution of online social networks. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 611–617. ACM, 2006.

36. Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A.: Trawling the web for emerging cyber-communities. Computer Networks (Amsterdam, Netherlands: 1999), 31(11-16):1481–1493, 1999.

37. Lahiri, M. and Berger-Wolf, T. Y.: Structure prediction in temporal networks using frequent subgraphs. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, pages 35–42, 04 2007.

38. Lahiri, M., Maiya, A. S., Caceres, R. S., Habiba, and Berger-Wolf, T. Y.: The impact of structural changes on predictions of diffusion in networks. In Workshops Proceedings of the 8th IEEE International Conference on Data Mining, pages 939–948, 12 2008.

39. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z.: Kronecker graphs: An approach to modeling networks. J. Mach. Learn. Res., 11:58, mar 2010.

40. Leskovec, J., Kleinberg, J., and Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In Proc. 11th ACM SIGKDD on Knowledge discovery in data mining, pages 177–187, New York, NY, 2005. ACM.

41. Liben-Nowell, D. and Kleinberg, J.: The link prediction problem for social networks. In Proceedings of the twelfth international conference on Information and knowledge management, pages 556–559. ACM, 2003.

42. Meila, M.: Comparing clusterings: an axiomatic view. In In ICML '05: Proceedings of the 22nd international conference on Machine learning, pages 577–584. ACM Press, 2005.

43. Miller, B. A., Bliss, N. T., and Wolfe, P. J.: Toward signal processing theory for graphs and non-euclidean data. In ICASSP, pages 5414–5417, 2010.

44. Molloy, M. and Reed, B.: A critical point for random graphs with a given degree sequence. A critical point for random graphs with a given degree sequence, 6(2-3):161–180, 1995.

45. Molloy, M. and Reed, B.: The size of the giant component of a random graph with a given degree sequence. Comb. Probab. Comput., 7(3):295–305, 1998.

46. Moody, J., McFarland, D., and Bender-deMoll, S.: Dynamic network visualization. American Journal of Sociology, 110(4):1206–1241, 2005.

47. Morris, M. and Kretzschmar, M.: Concurrent partnerships and transmission dynamics in networks. Social Networks, 17:299–318, 1995.

48. Newman, M.: The structure and function of complex networks. SIAM Review, 45:167–256, 2003.

49. Papadimitriou, S., Li, F., Kollios, G., and Yu, P. S.: Time series compressibility and privacy. In In VLDB, pages 459–470, 2007.

50. Partridge, C., Cousins, D., Jackson, A. W., Krishnan, R., Saxena, T., and Strayer, W. T.: Using signal processing to analyze wireless data traffic. In Proc. ACM workshop on Wireless Security, pages 67–76, 2002.

51. Pesaran, M. H. and Timmermann, A.: Model instability and choice of observation window. Economics Working Paper Series 99-19, Department of Economics, UC San Diego, Sep 1999.

52. Riolo, C., Koopman, J., and Chick, S.: Methods and measures for the description of epidemiologic contact networks. Journal of Urban Health, 78:446–457, 2001.

53. Rissanen, J.: Modeling by shortest data description. Automatica, 14:465–471, 1978.

54. Rubenstein, D., Sundaresan, S., Fischhoff, and Saltz, D.: Social networks in wild asses: Comparing patterns and processes among populations, pages 159–176. Martin-Luther-University Halle-Wittenberg, Halle, 2007.

55. Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A.: CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31), Jan 2006.

56. Sedley, D.: The stoic criterion of identity. Phronesis, 27:255–75, 1982.

57. Shannon, C. E.: A mathematical theory of communication. The Bell System Technical Journal, 27:379–423, 623–656, 07 1948.

58. Shetty, J. and Adibi, J.: Enron email dataset. Technical report, Institute, USC Information Sciences, 2004.

59. Silk, J., Alberts, S., and Altmann, J.: Social relationships among adult female baboons (papio cynocephalus) ii. variation in the quality and stability of social bonds. Behavioral Ecology and Sociobiology, 61(2):197–204, 2006.

60. Sulo, R., Berger-Wolf, T., and Grossman, R.: Meaningful selection of temporal resolution for dynamic networks. In Proc. 8th Workshop on Mining and Learning with Graphs, MLG '10, pages 127–136, New York, NY, USA, 2010. ACM.

61. Sun, J., Faloutsos, C., Papadimitriou, S., and Yu, P. S.: Graphscope: parameter-free mining of large time-evolving graphs. In KDD '07: Proc. 13th ACM SIGKDD on Knowledge discovery and data mining, pages 687–696, New York, NY, USA, 2007. ACM.

62. Sundaresan, S. R., Fischhoff, I. R., Dushoff, J., and Rubenstein, D. I.: Network metrics reveal differences in social organization between two fission-fusion species, grevy's zebra and onager. Oecologia, September 2006.

63. Tanaka, Y., Iwamoto, K., and Uehara, K.: Discovery of time-series motif from multi-dimensional data based on mdl principle. Machine Learning, 58(2-3):269–300, February 2005.

64. Tantipathananandh, C., Berger-Wolf, T., and Kempe, D.: A framework for community identification in dynamic social networks. In KDD '07: Proc. 13th ACM SIGKDD on Knowledge discovery and data mining, pages 717–726, New York, NY, USA, 2007. ACM.

65. Wasserman, S. and Faust, K.: Social Network Analysis: Methods and Applications. Number 8 in Structural analysis in the social sciences. Cambridge University Press, 1 edition, 1994.

66. Yosef, N. and Regev, A.: Impulse control: temporal dynamics in gene transcription. Cell, 144:886–896, 2011.

VITA

Name

Rajmonda Sulo Cáceres

Education

Ph.D., Mathematics, University of Illinois at Chicago, Chicago, Illinois, 2012
M.S., Mathematics and Computer Science, University of Illinois at Chicago, Chicago, Illinois, 2005
B.S. (Hons.), Mathematics and Computer Science, University of Illinois at Chicago, Chicago, Illinois, 2002

Honors

National Science Foundation Travel Award, 2011
LAS PhD Travel Award, 2011
Women in Science and Engineering Travel Grant, 2008, 2010, 2011
National Center for Data Mining Research Grant, 2010
Graduate Student Council Travel Grant, 2008, 2010
MCS Department Travel Grant, 2008
Winner of the High Performance Analytical Challenge at Super Computing Conference, 2007
Grace Hopper Scholarship sponsored by the National Science Foundation, 2007
Dean's Honor List, 2000
President's Honor List, 1999

Professional Experience

Graduate Researcher

- Computational Population Biology Lab, University of Illinois at Chicago, 2008 - 2012
- Women in Science and Engineering Program, University of Illinois at Chicago, 2011- 2012
- National Center for Data Mining, University of Illinois at Chicago, 2005 - 2008
- Urban Transportation Center, University of Illinois at Chicago, 2002 - 2008

Adjunct Professor

- Dominican University, River Forest, IL, 2010 - 2011

Teaching Assistant

- Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 2004-2005, 2008-2010

Publications

Caceres, R., Berger-Wolf, T. and Grossman, R.: Temporal Scale of Processes in Dynamic Networks, IEEE ICDM 2011 Workshop on Data Mining in Networks, Vancouver, Canada, 2011.

Sulo, R., Berger-Wolf, T. and Grossman, R.: Meaningful selection of temporal resolution for dynamic networks, Workshop on Mining and Learning with Graphs, Washington DC, 2010.

Barale, C. L., Kulahci, I. G., Habiba, Sulo, R., Berger-Wolf, T. and Rubenstein, D.I.: A Social Networks Approach to Sheep Movement and Leadership, 7th International Conference on Applications of Social Network Analysis, Zurich, Switzerland, 2010.

Devarajan, K., Echeverry-Galvis, M.A., Sulo, R. and Peterson, J.K.: Unusual Relationships, Python and Weaver Birds, Proceedings of the 9th Python in Science Conference, Austin, Texas, 2010.

Lahiri, M., Maiya, A., Sulo, R., Habiba, and Berger-Wolf, T.: The Impact of Structural Changes on Predictions of Diffusion in Networks, ICDM Workshop on Analysis of Dynamic Networks, Pisa, Italy, 2008.

Thakuriah, P.,Yanos, G., Lin, J., Metaxatos, P., Sulo, R., Pu, W. and Mbekeani, L.: Including Weather Effects in Real-Time Traffic Information Environment, Exploratory Analysis and Alternative Models, Proceedings of Intelligent Transportation Systems World Congress, 2008.

Metaxatos, P., Sulo, R., Limber, J. and Yanos, G.: Development of Adaptive Technology for the University of Illinois at Chicago Shuttle Bus, International Conference on the Application of Advanced Technologies in Transportation, Athens, Greece, 2008.

Grossman,R. , Sabala, M., Gu, Y., Anand, A., Handley, M., Sulo, R. and Wilkinson, L.: Distributed Discovery in E-Science, Lessons from the Angle Project, Super Computing Conference, Reno, Nevada, 2007.

Sulo, R., Anand, A., Wilkinson, L., Grossman, R., and Eick, S.: Topographically-Based Real-Time Traffic Anomaly Detection in a Metropolitan Highway System, IEEE Visual Analytics Science and Technology, Baltimore, Maryland, 2006.

Sulo, R., Eick, S. and Grossman, R.: DaVis, A tool for Visualizing Data Quality, IEEE Symposium on Information Visualization, Minneapolis, Minnesota, 2005.