University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Nino Antulov-Fantulin

Statistical inference algorithms for epidemic processes on complex networks

DOCTORAL THESIS

Zagreb, 2015

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Nino Antulov-Fantulin

Statistical inference algorithms for epidemic processes on complex networks

DOCTORAL THESIS

Supervisors:
Associate Professor Mile Šikić, PhD
Tomislav Šmuc, PhD

Zagreb, 2015

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Nino Antulov-Fantulin

Algoritmi za statističko zaključivanje o epidemijskim procesima na kompleksnim mrežama

DOKTORSKI RAD

Mentori:
izv. prof. dr. sc. Mile Šikić
dr. sc. Tomislav Šmuc

Zagreb, 2015.

The dissertation was made at the University of Zagreb Faculty of Electrical Engineering and Computing, Department of Electronic Systems and Information Processing in cooperation with Ruđer Bošković Institute, Department of Electronics, Computational Biology and Bioinformatics Group.

Supervisors:

Associate Professor Mile Šikić, PhD

Tomislav Šmuc, PhD

The dissertation has: 107 pages

The dissertation number:

About the Supervisors

Mile Šikić received M.Sc degree in 2002 from the Faculty of Electrical Engineering in Zagreb. In 2008 he defended his PhD thesis "Computational method for prediction of protein-protein interaction" at the same faculty. He started his career at University of Zagreb in 1997 as research associate at Faculty of Electrical Engineering and Computing. He worked as teaching assistant in the period from 2005 to 2009, and as assistant professor from 2009-2014. From 2015 he has been working as associate professor. Using his annual leave he joined Bioinformatics Institute, A*STAR Singapore as postdoc in the period from 2011 to 2012. In 2013 he was appointed as adjunct scientist at the same institute. He has been working as consultant and project manager on tens of industry projects in the ICT field. Also he has been leading several research projects in bioinformatics and computational biology. He was a one of leading investigators on the project „Complex network analysis of cis and trans chromatin interactions" financed by JCO Singapore. In 2014 he received Proof of Concept BICRO grant, Croatian Science Foundation grant and a donation from ADRIS Foundation. In the fields of bioinformatics and complex networks he has authored or co-authored 10 journal papers (CC, SCI, SCI expanded), 6 papers in conference proceedings and one book chapter. His work has been cited over 200 times (h-index = 6). He teaches Software Engineering, Algorithms and data structures and bioinformatics. At University of Zagreb, he created and co-created new courses Bioinformatics and Creative Laboratory. He has supervised over 50 undergraduate and master thesis, 2 master dissertations, and currently is supervising three PhD students. He is a member of steering committee of Science and Society Synergy Institute - ISZD.

Tomislav Šmuc received M.Sc degree in 1991 from the Faculty of Electrical Engineering in Zagreb. In 1994 he defended his PhD thesis "Improving stochastic optimization methods for optimization schemes replacing fuel reactors with water under pressure" at the same faculty. In the period from 1986-1998 he participates in several research projects of the Ministry of Science of Croatia and CRP (Coordinated Research Project) of International Atomic Energy Agency (IAEA) related to in-core fuel management of nuclear reactors, simultaneously working on long-term loading pattern optimization projects with the Nuclear Power Plant Krško. In that period he has published a considerable number of research papers and technical reports, and has also designed several software packages for in-core fuel management calculations of nuclear reactors. Since 1999 the focus of his research work is in the field of artificial intelligence, data mining and machine learning, with emphasis on applications in bio-medicine , engineering and economics. Since 2006, the focus of his research is the application of modern machine learning techniques in molecular biology (genomics and proteomics), determination of gene/protein function, modeling of the structure and activity relationship of chemical substances, and improving the interpretation of the experimental data in proteomics. Despite the focus of the application in the field of life sciences, he continues to work on the application of machine learning techniques in the fields of economics and social sciences and of engineering (meta- modeling). Since 2006 participated in several European projects, either as a collaborator or work package leader (HEARTFAID , e - LICO, FOC, MultiPlex , MAESTRA ; InnoMol). He has authored or co-authored more than 60 articles in scientific journals or in proceedings of international conferences (SCOPUS: h-index=9; citations=372).

O mentorima

Mile Šikić je 1996. godine završio diplomski studij, magistrirao 2002, a doktorirao 2008 s doktorskom disertacijom pod naslovom „Računalna metoda za predviđanje mjesta proteinskih interakcija“ na Fakultetu elektrotehnike i računarstva, Sveučilišta u Zagrebu. Od 1997 do 2005 je radio kao zavodski suradnik na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu, od 2005 do 2009 kao asistent, 2009 izabran je u zvanje docenta, a 2015 u izvanrednoga profesora. Od 1.5.2011 – 1.9.2002 kao poslijedoktorand je radio na Bioinformatics Institut, A*STAR u Singapuru, a od 2013 na istom institutu djeluje kao pridruženi znanstvenik. Sudjelovao je i vodio nekoliko desetaka projekata s industrijom u području ICT-a te nekoliko istraživačkih projekata primarno u području bioinformatike i računalne biologije. Kao jedan od vodećih istraživača aktivno je sudjelovao na projektu „Complex network analysis of cis and trans chromatin interactions” financiranim od strane JCO Singapur. Voditelj je HRZZ uspostavnoga projekta te Proof of Concept BICRO projekta. Projekti koje je vodio su dobili donacije Zaklade Adris i Zaklade HAZU. Do sada je, većinom u području bioinformatike i računalne biologije, objavio deset znanstvenih radova (CC, SCI, SCI expanded), jedno poglavlje u knjizi i šest radova na konferencijama s međunarodnom recenzijom. Njegovi radovi su do sada citirani 200 puta. Predaje na prediplomskoj i diplomskoj razini predmete vezane uz programiranje, algoritme i strukture podataka, bioinformatiku i kreativnost. U nastavu je na diplomskoj razini uveo predmete Bioinformatika te Kreativni laboratorij koji je pokrenuo zajedno s još 5 kolega s različitih sastavnica Sveučilišta u Zagrebu. Vodio je više od 50 završnih i diplomskih radova, dva magistarska rada, a trenutno je mentor trojici doktorskih studenata. Član je upravnoga vijeća Instituta sinergije znanosti i društva.

Tomislav Šmuc diplomirao je 1986. godine na Elektrotehničkom fakultetu u Zagrebu. Godine 1994. obranio je doktorsku disertaciju " Poboljšanje stohastičkih metoda optimizacije za programe za optimizaciju zamjene reaktora goriva s vodom pod pritiskom " na istom fakultetu. U razdoblju od 1986-1998 sudjeluje u nekoliko znanstvenih projekata Ministarstva znanosti Republike Hrvatske i CRP projekata (Coordinated Research Project) Međunarodne agencije za atomsku energiju (IAEA) u području gospodarenja gorivom u jezgri nuklearnih reaktora. Istovremeno radi na dugoročnom projektu s NE KRŠKO na dizajnu shema zamjene goriva. U tom razdoblju je objavio veći broj znanstvenih radova i tehničkih izvješća, kao i nekoliko programskih paketa za proračune shema zamjene goriva u nuklearnim reaktorima. Od 1999. godine fokus njegovog istraživačkog rada je u području umjetne inteligencije, dubinskoj analizi podataka i strojnom učenju, s naglaskom na primjenama u bio-medicini, inženjerstvu i ekonomiji. Od 2006. godine, fokus njegovog istraživanja je primjena suvremenih tehnika strojnog učenja u molekularnoj biologiji (genomika i proteomika): određivanje funkcije gena / proteina, modeliranje strukture i aktivnosti kemijskih spojeva, te u poboljšanju interpretacije esperimentalnih podaci u proteomike. Unatoč fokusu primjene na područja bio-znanosti i dalje nastavlja raditi na primjeni tehnika strojnog učenja i u drugim područjima: u području ekonomskih i društvenih znanosti kao i u području inženjerstva (meta modeliranje) . Od 2006. sudjelovao je u nekoliko europskih projekata HEARTFAID, e - LICO, FOC, MultiPlex , Maestra, InnoMol), kao suradnik ili voditelj radnog paketa ("workpackage leader"). Autor je ili ko-autor preko 60 publikacija u međunarodnim časopisima odnosno zbornicima međunarodnih znanstvenih skupova (SCOPUS: h-index=9; broj citata=372).

Acknowledgement

At the start, I would like to thank my supervisors and other colleagues who have helped me to finish this Ph.D. thesis. My supervisors Mile Šikić and Tomislav Šmuc have been more than supportive and always have had a lot of time for discussions, advices and help to progress my thesis contributions in computer science. At the same time, I would like to give a big thank to my fellow colleague Alen Lančić who has been my informal mathematical advisor and friend during many years. A person which introduced me to the field of complex networks and epidemics during my high school times is a theoretical physicist Hrvoje Štefančić, which was also my informal advisor for studying epidemic processes during my PhD. I would also like to thank to physicist Vinko Zlatić from Rudjer Boskovic Institute, with whom I have had a lot of discussions about complex networks. I would like to thank Prof. Yaneer Bar-Yam from the New England Complex Systems Institute for valuable discussions about the multiple source recognition in the early stages of research and Prof. Dirk Brockmann from the Robert Koch Institute for valuable insights about the epidemic modelling and source detection during my research internship. Many thanks to all my friends, which made things easier and funnier.

In the end, I would like to thank to my family Antulov-Fantulin (Darko, Azra, Marija, Bruno, Dunja) who have always been the great support and who enabled me to be here today.

∞

La vita è bella

Abstract

The main topics of this dissertation are novel methods and algorithms for the modelling and the statistical inference about epidemic processes based on the Susceptible-Infected-Recovered (SIR) model on arbitrary network structures. Two types of problems are solved: (i) estimation of the final epidemic outcome ("forward in time" statistical estimate) and (ii) estimation of epidemic initial conditions from a single epidemic realization ("backward in time" statistical inference). In order to estimate the final epidemic outcome on arbitrary networks without following the temporal dynamics, a novel FastSIR algorithm is constructed. The FastSIR algorithm is using a probability distribution of the number of infected nodes in a first neighbourhood in a limit of time to speed up the simulation. In the backward statistical inference, we solve two problems: (a) the detection of a single epidemic source from a realization and (b) the recognition that a realization has multiple initial sources. A number of different statistical estimators are presented for determining the likelihood for potential source producing the observed epidemic realization. The estimates are based on the Monte Carlo simulations of an epidemic spreading process on a network from a set of potential source candidates, which were infected in the observed realization. This statistical inference framework is applicable to arbitrary networks and different dynamical spreading processes. The problem of multiple-source epidemic recognition from a single realization is solved by constructing a statistical outlier detection algorithm, which is based on the Kolmogorov-Smirnov statistics over realization similarity distributions.

Key words: complex networks, epidemic spreading algorithms, statistical inference

Produženi sažetak

Algoritmi za statističko zaključivanje o epidemijskim procesima na kompleksnim mrežama

Glavna tema ove disertacije su nove metode i algoritmi za modeliranje i statističku procjenu epidemijskih procesa bazirani na stohastičkom modelu Podložan-Zaražen-Oporavljen (engl. Susceptible-Infected-Recovered - SIR), na proizvoljnim mrežnim strukturama. Algoritmi i metode posvećeni su rješavanju dva tipa problema: (i) procjena ishoda epidemije ("unaprijed u vremenu") te (ii) procjena početnih uvjeta epidemije iz jedne realizacije epidemijskog procesa ("unatrag u vremenu").

Podložan-Zaražen-Oporavljen (SIR) model parametriziran je s dva osnovna parametra: parametar p - vjerojatnost da zaraženi čvor prenese zarazu na podložnog susjednog čvora u mreži u diskretnoj jedinici vremena i parametar q - vjerojatnost da zaraženi čvor se oporavi u diskretnoj jedinici vremena. Za procjenu ishoda epidemije na mrežnoj strukturi konstruiran je osnovni Naive SIR algoritam. Naive SIR algoritam slijedi vremensku dinamiku stohastičkog procesa na mreži koristeći Monte-Carlo simulaciju. Isti algoritam koristi sljedeće strukture podataka: lista susjedstva - za konstantni pristup susjedima u mreži, struktura red - za pohranu zaraženih čvorova te indikatorsku strukturu polja za provjeru podložnosti i oporavljenosti čvora. Vremenska složenost Naive SIR algoritma proporcionalna je umnošku prosječnog broja zaraženih čvorova, prosječnom stupnju čvora te prosječnom vremenu oporavka. Na regularnim m-arnim stablima moguće je dobiti analitičku gornju granicu za prosječno vremensku složenost. Radi ubrzanja vremenske složenosti Naive SIR algoritma konstruiran je FastSIR algoritam koji ne slijedi vremensku dinamiku epidemije. FastSIR algoritam koristi distribuciju vjerojatnosti broja zaraženih u prvom susjedstvu u limesu vremena kako bi ubrzao simulaciju. Vremenska složenost FastSIR algoritma proporcionalna je umnošku prosječnog broja zaraženih čvorova i prosječnom stupnju čvora. Rekurzivna definicija za računanje distribucije vjerojatnosti broja zaraženih u prvom susjedstvu u limesu vremena je dana u disertaciji. Eksperimenti na realnim i sintetskim mrežama pokazuju ubrzanje (proporcionalno prosječnom vremenu oporavka) FastSIR algoritma u odnosu na Naive SIR algoritam. Predložena je inačica FastSIR algoritma za neprekinuto vrijeme koristeći analitički izvod za distribuciju vjerojatnosti broja zaraženih u prvom susjedstvu u limesu vremena za SIR model u neprekinutom vremenu.

Procjenom unazad u vremenu rješavamo dva problema: (a) detekcija izvora zaraze i (b) prepoznavanje realizacija koje dolaze iz više izvora. U tezi je predstavljeno više različitih procjenitelja izglednosti da zaraza iz određenog izvora reproducira promatranu realizaciju. Estimatori se temelje na Monte Carlo simulacijama zaraze iz skupa potencijalnih izvora, koji

su bili zaraženi u promatranoj realizaciji i mogu se podijeliti na dvije kategorije: heuristički (Naive Bayes, AvgTopK, AUCDF) i aproksimacijski (Direct Monte-Carlo, Soft Margin) estimatori. Naive Bayes estimator pretpostavlja nezavisnost između čvorova što omogućuje faktorizaciju združene vjerojatnosti realizacije na umnožak vjerojatnosti stanja čvorova u promatranoj realizaciji. Algoritmi AUCDF i AvgTopK estimiraju izglednost izvora na temelju empirijske kumulativne distribucije sličnosti između promatrane realizacije i realizacija generiranih iz potencijalnih izvora. Direct Monte-Carlo algoritam estimira izglednost izvora na temelju frekvencije generiranja promatrane realizacije iz određenog izvora koristeći tehniku podrezivanja krivo generiranih realizacija pomoću Monte-Carlo metode. Važno je napomenuti da tehnika podrezivanja ne unosi grešku u estimaciju izglednosti izvora. Soft Margin estimator relaksira kriterij prihvaćanja realizacija koristeći težinsku funkciju nad sličnostima generiranih realizacija. Soft Margin estimator prelazi u Direct Monte-Carlo estimator kada širina težinske funkcije teži u nulu. Testiranje na sintetskim mrežama pokazuje da Soft Margin estimator ima najbolje karakteristike (točnost rangiranja izvora i procjena izglednosti izvora) u odnosu na druge konkurentske estimatore iz literature. Također je pokazana primjenjivost Soft Margin estimatora u određivanju izvora zaraze na simuliranim epidemijama na realnim mrežama: temporalna mreža seksualnih kontakata i težinska mreža avionskih letova. Soft Margin estimator također omogućuje relaksaciju poznavanja parametara epidemije, strukture mreže ili stanja svih čvorova.

Sustavno prepoznavanje realizacija iz više mogućih izvora dovodi do eksponencijalnog porasta broja članova za analizu. Stoga se problem prepoznavanja realizacija koje dolaze iz više izvora prebacio na problem detekcije statističkih iznimki. Realizacije koje dolaze iz više izvora smatraju se iznimkama s obzirom na skup realizacija koje dolaze iz jednog izvora. Konstruiran je algoritam koji koristi Kolmogorov-Smirnov statistiku nad distribucijama sličnosti realizacija jednog prema jednom izvoru te sličnosti između promatrane realizacije prema realizacijama iz jednog izvora. Radi smanjenja vremenske složenosti implementirane su sljedeće optimizacijske tehnike: paralelizacija algoritma, poduzorkovanje potencijalnih izvora, kompresija realizacija u skup cjelobrojnih vrijednosti te brzo binarno računanje sličnosti. Metoda prepoznavanja realizacija iz više izvora testirana je na simuliranim zarazama na mrežama.

Ključne riječi: kompleksne mreže, algoritmi za širenje epidemija, statistička procjena

Contents

1. Introduction . . . 1

1.1. Objectives . . . 2

1.2. Overview of epidemic modelling research . . . 2

1.3. Structure of the thesis . . . 5

2. Forward in time epidemic modelling . . . 7

2.1. Problem formulation . . . 7

2.2. The Naive SIR algorithm . . . 7

2.3. Discrete FastSIR algorithm . . . 10

2.4. The FastSIR experiments . . . 20

2.5. Continuous FastSIR algorithm . . . 23

3. Backward in time - single source epidemic detection . . . 27

3.1. Problem formulation . . . 27

3.2. The AUCDF and the AvgTopk – heuristic similarity-based methods . . . 29

3.3. The Naive Bayes – heuristic independence-based method . . . 32

3.4. The Direct Monte Carlo – approximate method . . . 32

3.5. The Soft Margin – approximate relaxed method . . . 36

3.6. Benchmark analysis . . . 40

3.7. Time complexity of estimators . . . 45

3.8. Empirical temporal network of sexual contacts – a case study . . . 46

3.9. Empirical weighted network of air traffic – a case study . . . 48

4. Backward in time - multiple source epidemic recognition . . . 53

4.1. Problem formulation . . . 53

4.2. Statistical properties of 2-source epidemic processes . . . 54

4.3. Outlier detection method . . . 58

4.4. Results . . . 63

5. Conclusion and discussion . . . 71

Appendices 73

A. Fundamentals of complex network theory 73

B. Fundamentals of epidemic modelling on networks 83

C. Alternative SIR algorithms 91

Bibliography 95

Biography 103

Životopis 107

Chapter 1

Introduction

Through the history, epidemic outbreaks like the Black Death in 1346, the Cholera in 1817, the Spanish Flu in 1918 or more recent outbreaks like the SARS in 2002, the H1N1 in 2009 and the EBOLA in 2014 had a large impact on the human society. Therefore, it is reasonable that scientists for decades have tried to model, predict and control the epidemics in the human population. Over the last 15 years, it has been discovered that the structure of the vast majority of biological, technological and social systems can be represented by complex networks. These findings established a new interdisciplinary field of complex networks [1, 2, 3], which gave us new insights for epidemic modelling [4]. Detailed information about the fundamentals of the complex networks theory are given in the Appendix A.

Different approaches to epidemic modelling exist, but this thesis concentrates on the computational epidemic modelling of the SIR model on arbitrary network structures. The SIR model is a cornerstone model for many infectious diseases, where each individual in a population can be in one of the three different compartments. Those who are susceptible to the disease are in the Susceptible compartment, those who are infected and can transmit the disease to others are in the Infected compartment and those who have recovered and are immune and those who are removed from the population are in the Recovered compartment. Although the SIR compartment model [5] was originally constructed for disease modelling in human population, its variants can be used to study computer virus propagation [6] in computer networks or information and rumour propagation [7] in social networks. Some infectious diseases are described with models that have a different number of compartments (SI, SIS, SEIR, ...), e.g. SIS model (Susceptible-Infected-Susceptible) in which individuals do not have long lasting immunity and therefore the Recovered compartment does not exist.

1.1 Objectives

The thesis is based on two main research objectives: formulating (i) forward in time and (ii) backward in time statistical inference algorithms about stochastic epidemic processes on network structures. The forward in time modelling estimates the final outcome of an epidemic spreading for a given initial condition. On the contrary, in the backward in time objective, a statistical estimate of the initial conditions for a given epidemic realization is given.

The three research questions of this thesis are:

1. Is it possible to calculate the final outcome of the discrete SIR stochastic epidemic process for a given initial condition on an arbitrary network structure with a procedure which neglects epidemic dynamics?
2. Can we detect the source location of an epidemic realization spread over a network structure under the SIR epidemic model?
3. Can we detect that the spread of disease is a result of a single or multiple source propagation over a network structure under the SIR model?

Along with answering these research questions, three corresponding scientific contributions are:

1. Construction of a novel fast algorithm for determining the outcome of a stochastic epidemic process based on the Susceptible-Infected-Recovered (SIR) model on arbitrary network structure.
2. Statistical inference methods for epidemic source detection from a single realization of a stochastic epidemic process based on the SIR model on arbitrary network structure.
3. Statistical inference methods for discriminating between single and multiple-source realizations of a stochastic epidemic process based on the SIR model on arbitrary network structure.

1.2 Overview of epidemic modelling research

The field of the classical mathematical epidemiology has a long history in modelling the epidemic processes [8] by using: (a) the set of ordinary differential equations for macro-level dynamics description of deterministic processes, (b) the Markov chain theory for micro-level dynamics description of stochastic processes and (c) the stochastic differential equations for macro-level dynamics description of stochastic processes.

Another important class of epidemic modelling is modelling on the network structures [9], where we have: (A) models on the contact networks where nodes represent individuals and edges represent contacts between the individuals and (B) the meta-population models where nodes represent populations and edges represent travelling connections.

In this thesis, we are interested in modelling the stochastic epidemic processes on the contact network structures, for which different approaches exist: (i) the bond percolation approach, (ii) the mean field approach, (iii) the message passing approach and (iv) the computational approach. Detailed information about the theoretical fundamentals of epidemic modelling on contact networks are given in the Appendix B.

Depending on the assumptions that the approaches make, we divide them into two big categories: the homogeneous mixing framework and the heterogeneous mixing framework. The homogeneous mixing framework assumes that all individuals in a population have an equal probability of contact. This is a traditional mathematical framework [5], [10], where differential equations can be applied to understand epidemic dynamics. The models in this framework predict the epidemic threshold which divides the nonspreading and the spreading phase of the SIR and the SIS models. In reality, each individual has a contact with only a small fraction of individuals in a population. As the assumption of the homogeneous mixing fails to describe the realistic scenario of disease spreading, the heterogeneous mixing is introduced by using a network structure. The small world network property [11] and the scale-free network property [12] [13] observed in empirical networks have a great impact on the outcome of epidemic spreading. Same models in the heterogeneous mixing framework predict that there is no non-zero epidemic threshold for a certain power-law degree distribution for the SIR and the SIS epidemic models, which implies that a disease will always spread [14]. The bond percolation approach applies the percolation theory to describe the epidemic processes on networks [15], [16], [17]. The percolation theory predicts the mean epidemic size, but neglects epidemic dynamics. Analytical solutions of a mean outbreak size of the configuration network models have also been derived [18]. In order to show isomorphism of epidemic models to bond percolation processes [19], the epidemic percolation networks were introduced as a valuable tool to study stochastic epidemic models. The Monte Carlo algorithms for fast bond percolation [20] have also been formulated. The mean-field approach assumes that all nodes in a network with the same degree k with respect to an epidemic process are statistically equivalent [21], [14]. This method enables us to write the epidemic time evolution equations for a network with an arbitrary degree distribution. By solving them, relations of topology dependent features and the epidemic threshold have been discovered [22]. However, the stochastic fluctuations and finite sizes can play a crucial role in the final epidemic outcome, e.g. the individual-based Monte Carlo simulations show that the extent of disease spreading is in general characterized by a bimodal probability distribution [23]. The general epidemic model can also be mapped to the time-dependent message passing [24, 25] on the contact network, which is exact on trees and locally tree-like networks in system size limit and gives a bound on on-tree-like networks. For binary-state dynamics on the configuration network the approximate master equations were developed [26], which generalize the pair-approximation and mean field method .

The particle network approach is a meta-population model, which assumes that individuals are represented by particles which diffuse along the edges of a network and each node contains some non-negative integer number of particles (reaction-diffusion process simulations [27]). Some studies [28] used the contact network models between urban cities (cities are connected through the airline transportation network) and the homogeneous mixing model inside urban cities and examined the influence of interventions (antiviral drugs and containments) to a spread of a pandemic. Realistic computational epidemic simulations (GLEaMviz [29], EpiFast [30], EpiSims [31] and EpiSindemics [32]) have become a very important application of high-performance computing in epidemic predictions. These are the most important examples of realistic epidemic simulations that can be used in public health studies.

Although the history of epidemic modelling started around 1930, the inverse problem of estimating the initial conditions like a patient-zero on networks has only recently be formulated. This has led to the development of a number of different source detection estimators for static networks, which vary in their assumptions on the network structure and the spreading process models [25, 33, 34, 35, 36, 37, 38, 39, 40, 41]. For the source detection with the SI model (Susceptible-Infected) the following interesting results have been obtained. Zaman et. al. developed a rumour centrality measure, which is the maximum likelihood estimator for regular trees under the SI model [33]. Dong et. al. Also studied the problem of rooting the rumour source with the SI model and demonstrated the asymptotic source detection probability on regular tree-type networks [34]. Comin et. al. compared the different centrality measures, e.g. the degree, the betweenness, the closeness and the eigenvector centrality as the source detection estimators [40]. Wang et. al. addressed the problem of source estimation from multiple observations under the SI model [35]. Pinto et. al. used the SI model and assumed that the direction and the times of the infection are known exactly, and solved diffusion tree problem using breadth first search from sparsely placed observers [37]. In the case of the SIR model (Susceptible-Infected-Recovered) there are two different approaches. Zhu et. al. adopted the SIR model and proposed a sample path counting approach for the source detection [36]. They proved that the source node on infinite trees minimizes the maximum distance (Jordan centrality) to the infected nodes. Lokhov et. al. used the dynamic message-passing algorithm (DMP) for the SIR model to estimate the probability that a given node produces the observed snapshot. They use a mean-field-like approximation (independence approximation) and an assumption of a treelike contact network to compute the marginal probabilities [25]. Altarelli. et. al. remove the independence assumption and use the message passing method with an assumption of a treelike contact network to estimate the source [38].

Prakash et. al. use the Minimum Length Description principle to find a set of nodes that best explain the snapshot under the SI model on networks with NetSleuth algorithm [42]. Zang et. al. construct a score-based reverse propagation algorithm for SIR model by using different centrality measures to find out recovered nodes from susceptible ones and then employ community detection algorithms to resolve multi-source problem (each source is mapped to one community) [43]. Chen et. al. study the problem of detecting multiple sources with the SIR model by developing a sample-path-based algorithms for tree structures and propose a generalization for general networks [44].

1.3 Structure of the thesis

The Chapter 2, describes the Forward in time epidemic modelling on arbitrary networks using the Monte Carlo algorithms. In this chapter, we formulate the forward in time modelling problem (section 2.1) and then present two algorithms which solve this problem. The first algorithm is the Naive SIR algorithm (baseline) in section 2.2, which follows natural epidemic dynamics in time, but uses efficient data structures. Then, the FastSIR algorithm is presented in section 2.3, it is the algorithm for the discrete SIR model which does not follow epidemic dynamics in time, but improves the running time complexity of the Naive SIR algorithm. A series of experiments and running time measurements for the Naive SIR and FastSIR algorithms are presented in section 2.4. Than, the continuous time SIR infection probability distribution is introduced in section 2.5.

Chapter 3 describes the backward in time inference of the epidemic source. In section 3.1 the epidemic source detection problem is formulated and in the following sections 3.2–3.5 different algorithms are presented (AUCDF, AvgTopK, Naive Bayes, Direct Monte Carlo and Soft Margin). In section 3.6, all the methods along with other state-of-the art methods are compared on a set of benchmark cases. The estimators time complexity is given in section 3.7. The case studies on empirical temporal network and world airport weighted network are given in sections 3.8 and 3.9.

Chapter 4 describes the backward in time inference of the multiple epidemic sources. The problem of recognizing the multiple source epidemic spread from the observed realization is formulated in section 4.1. In section 4.2 the statistical properties of 2-source epidemics are discussed, section 4.3 describes an algorithm for detecting multiple source processes as an outlier process with respect to the single source processes and section 4.4 gives the performance of an outlier detection method in experiments on synthetic networks.

Chapter 5 gives a conclusion of the thesis and in the Appendix section A and B, the fundamentals of network theory and epidemic modelling on networks are given.

Chapter 2

Forward in time epidemic modelling

2.1 Problem formulation

We define the contact-network as an undirected and non-weighted graph $G(N,L)$ (N-set of nodes, L-set of links). A link (u,v) exists only if two nodes u and v are in contact during the epidemic time. In this chapter, we also assume that the contact-network during the epidemic process is a static one. To simulate epidemic propagation through a contact-network, we use the standard stochastic SIR model. In this model each node at any time can be in one of the following states: susceptible (S), infected (I) or recovered (R). In this thesis, the discrete time SIR model will be used, although some results also extend to the continuous time SIR model (see Section 2.5). In epidemic modelling, we can either study dynamic (Naive SIR algorithm) or asymptotic properties (Naive SIR algorithm and FastSIR algorithm). Here, we are interested in finding a fast algorithm for inferring the asymptotic properties: e.g. node state probabilities or the expected outbreak size at the end of the epidemic process on an arbitrary network structure. Note, that by using techniques from statistical mechanics [45] a lot of approximate methods to estimate different properties of epidemic processes have been developed (see Appendix B), but these approaches neglect certain correlations like loops in network structures or dependencies among node states. In this thesis, we are interested in finding both accurate and fast statistical algorithms for epidemic processes on arbitrary networks.

2.2 The Naive SIR algorithm

I this algorithm, time is modelled in discrete time steps, and number of time steps necessary for one epidemic simulation is determined by the step at which epidemics stop spreading, i.e. when there are no infected nodes in the network. At the beginning of each epidemic simulation all nodes from graph G are in the susceptible state except an arbitrary set of nodes, which are initially infected. Infection process is characterized by epidemic parameters p and q, which together with the initially infected nodes represent initial conditions denoted by λ. The epidemic parameter p is the probability that an infected node u infects an adjacent susceptible node v in one discrete time step. The epidemic parameter q is the probability that an infected node recovers in one discrete time step. At the end of an epidemic simulation, all nodes can be in one of two following states: susceptible or recovered. Therefore, if some nodes got infected during the simulation process, they will certainly recover in the limit of time when the epidemic parameter q is non-zero.

In standard algorithm for SIR model, an infected node tries to infect its neighbours sequentially. For each neighbouring node a pseudo random number between 0 and 1 is calculated. If the number is smaller or equal to p value, the neighbouring node is infected. Then, we check if the node recovers according to a new pseudo random number and q parameter. Here, we call this algorithm the Naive SIR algorithm [46].

In the implementation (see Algorithm 1) the set of infected nodes is represented with a queue data structure I and susceptible nodes as an array structure S. If the array value of particular node is ”1” that node is susceptible. Vice versa, the node is infected or recovered. The network is represented using an adjacency list.

Algorithm 1 The Naive SIR algorithm

```
Input: $(G, \lambda)$ where $G$ is contact network and $\lambda$ represents the initial conditions. Initial conditions consist of $p$, $q$, $I$ a queue of initially infected nodes and $S(v)$ is an array indicator of susceptible nodes.
Output: array indicator of recovered nodes $R(v)$
while $I$ is not empty do
    dequeue($u$, $I$)
    for each contact $v$ of node $u$ do
        if $S(v)$ is equal to 1 then
            let the transmission of infection $u \rightarrow v$ occur with probability $p$
            if $u \rightarrow v$ does occur then
                update $S(v)$ and $R(v)$
                enqueue ($v$, $I$)
            end if
        end if
    end for
    update state of $u$ from infected to recovered with probability $q$
    if $u$ is not recovered then
        enqueue($u$, $I$)
    end if
end while
output $R(v)$
```

Time and space complexity analysis of the Naive SIR algorithm

Here, we examine the average case running time and space complexity of the Naive SIR algorithm. For the order of growth of the average case running time algorithm analysis, we use standard big-O notation (asymptotic upper bound within a constant factor) [47].

The average case running time of the Naive SIR algorithm $\overline{T_c}(\mathbb{E}[X], \overline{k}, q)$ is equal to:

$$\overline{T_c}(\mathbb{E}[X], \overline{k}, q) = O\left(\frac{\mathbb{E}[X]\overline{k}}{q}\right) \tag{2.1}$$

Where $\mathbb{E}[X]$ denotes total expected number of infected nodes and $\overline{k}$ denotes the average degree.

To explain this expression, let us start with the case of one infected node with k neighbours. In one cycle it tries to transmit infection to each of its neighbours. The run-time calculation cost of that is proportional to k. At the end of each cycle a random number is compared with q. If the number is greater than q the node is moved to set of recovered nodes. Total running time cost T_c^i for some infected node v_i is the sum of costs over all time steps where node v_i was infected. Hence, it can be seen that the number of cycles the node is in infected state is a sample from a geometric distribution with expectation $1/q$. Because of that, total average running time for one infected node is $\overline{T_c^1} = O(k/q)$. Let $\mathbb{E}[X]$ be the expected number of infected nodes in the network. Because the main while loop of the Naive SIR algorithm executes sequentially total average case running time $\overline{T_c}$ is the sum of $\overline{T_c^i}$ for all infected nodes v_i. The sum $\overline{T_c} = \overline{T_c^1} + \overline{T_c^2} + ... + \overline{T_c^n}$ has $\mathbb{E}[X]$ terms. Therefore, the average case running time $\overline{T_c}$ is $O(\mathbb{E}[X]\overline{k}\frac{1}{q})$.

For a network with cycles, it is difficult to analytically calculate the expected number of infected nodes, but we can calculate it for a regular m-arry tree. To that end, we will use X_n, random variable of a number of directly infected susceptible nodes by the infected node of degree n. It can be easily verified that $\mathbb{E}[X_n] = n\mathbb{E}[X_1] = n\mathbb{P}(X_1 = 1)$.

Proposition 2.2.1. *The average case running time of the Naive SIR algorithm* $\overline{T_c}(\mathbb{E}[T_n], \overline{k})$ *for a m-arry tree of depth n is equal to:*

$$\overline{T_c}(\mathbb{E}[T_n], \overline{k}) = O(\mathbb{E}[T_n]\overline{k}),$$

where T_n *is a random variable that measures time needed for epidemic to stop spreading in regular m-arry tree of depth n and* $\overline{k}$ *denotes the average degree.*

In particular, the expectation of T_n *satisfies the relation:*

$$\mathbb{E}[T_n] \leqslant \frac{1}{q}\frac{[m\mathbb{P}(X_1 = 1)]^n - 1}{m\mathbb{P}(X_1 = 1) - 1}$$

where the expression $\frac{[m\mathbb{P}(X_1=1)]^n-1}{m\mathbb{P}(X_1=1)-1} = \mathbb{E}[X]$ *is the expected total number of infected nodes [23].*

Proof.

$$\begin{aligned}\mathbb{E}\left[T_n\right] = \sum_{i=0}^{m} \mathbb{E}\left[T_n \,|X_m = i\right] \mathbb{P}\left(X_m = i\right) \leqslant \mathbb{E}\left[T_0\right] \mathbb{P}\left(X_m = 0\right) + \\ + \sum_{i=1}^{m} \left(\mathbb{E}\left[T_0\right] + \mathbb{E}\left[\max_{1 \leqslant j \leqslant i} T_{n-1}^{(j)} \right] \right) \mathbb{P}\left(X_m = i\right) = \\ = \frac{1}{q} + \sum_{i=1}^{m} \mathbb{E}\left[\max_{1 \leqslant j \leqslant i} T_{n-1}^{(j)} \right] \mathbb{P}\left(X_m = i\right) \leqslant \frac{1}{q} + \sum_{i=1}^{m} \mathbb{E}\left[\sum_{j=1}^{i} T_{n-1}^{(j)} \right] \mathbb{P}\left(X_m = i\right) = \\ = \frac{1}{q} + \mathbb{E}\left[T_{n-1}\right] \sum_{i=1}^{m} i \cdot \mathbb{P}\left(X_m = i\right) = \frac{1}{q} + \mathbb{E}\left[T_{n-1}\right] \mathbb{E}\left[X_m\right] = \\ = \frac{1}{q} + m\mathbb{E}\left[T_{n-1}\right] \mathbb{P}\left(X_1 = 1\right) \Rightarrow \mathbb{E}\left[T_n\right] \leqslant \frac{1}{q} \frac{\left[m\mathbb{P}\left(X_1 = 1\right)\right]^n - 1}{m\mathbb{P}\left(X_1 = 1\right) - 1}\end{aligned}$$

□

The space complexity S of the Naive SIR algorithm with respect to the number of links L and the number of nodes N is equal to:

$$S[L,N] \approx \underbrace{2L}_{G} + \underbrace{N}_{I} + \underbrace{N}_{S(v)} + \underbrace{N}_{R(v)} = 2L + 3N,$$

where the first term denotes the space complexity of contact network G (adjacency list), The second term denotes the space complexity of a queue of infected nodes I, the third term denotes the space complexity of an array indicator of susceptible nodes $S(v)$ and the last term denotes the space complexity of an array indicator of recovered nodes $R(v)$. Note, that $S(v)$ and $R(v)$ can be implemented as a bitset structure to further reduce memory consumption.

In connected networks $L \geqslant N$ and then the space complexity S of the Naive SIR algorithm is:

$$S[L,N] = O(L).$$

2.3 Discrete FastSIR algorithm

The main goal of the forward in time modelling is to find a faster algorithm for determining the node state at the end of epidemics, without following epidemic dynamics explicitly in time. Looking at the complexity of Algorithm 1, we can see that possible speedup of the sequential version of the algorithm can be obtained by reducing the $1/q$ part. Since we know how to calculate the probability distributions for the number of infected nodes [23], the idea is to choose that number from the distribution.

Proposition 2.3.1. *The probability that the infected node infects k neighbours out of total n*

susceptible neighbours in the limit of the discrete time under the SIR model (p,q) is [23]:

$$\mathbb{P}(X_n = k) = q\binom{n}{k}\sum_{l=0}^{k}\binom{k}{l}(-1)^l \frac{(1-p)^{n-k+l}}{1-(1-q)(1-p)^{n-k+l}}. \tag{2.2}$$

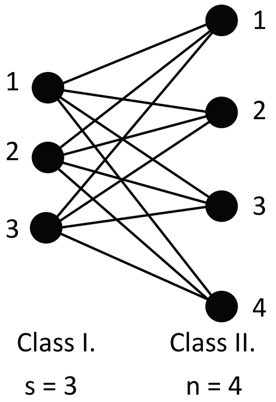


Figure 2.1: Visualization of undirected bipartite graph consisting of initially infected $s = 3$ nodes (class I) and initially susceptible $n = 4$ nodes (class II). Each node from the class I is connected with all nodes from the class II.

Proof. Let us consider an undirected bipartite graph consisting of initially infected nodes (class I) and initially susceptible nodes (class II). Each node from the class I is connected with all nodes from the class II (see Figure 2.1). Let A_i be an event that a node a_i from the class II gets infected, for $i \in \{1,\dots,n\}$ and 1_j be the associated indicator random variable. Furthermore, let $T_j, j \in \{1,\dots,s\}$ be random variables of recovery time for the nodes from the class I. We have

$$\mathbb{P}\left(1_i = 0, \sum_{j=1}^{s} T_j = m\right) = (1-p)^{s+m}\begin{pmatrix} m+s-1 \\ m \end{pmatrix} q^s(1-q)^m$$

Because $\sum_{j=1}^{s} T_j$ is a sum of s i.i.d. geometric random variables with the parameter q and as such has the negative binomial distribution with parameters s and q . Furthermore,

$$\mathbb{P}\left(\sum_{j=1}^{s} T_j = m, \sum_{i=1}^{n} 1_i = k\right) = \begin{pmatrix} n \\ k \end{pmatrix}\left(1-(1-p)^{m+s}\right)^k\left((1-p)^{m+s}\right)^{n-k}\begin{pmatrix} m+s-1 \\ m \end{pmatrix} q^s(1-q)^m$$

because $\sum_{i=1}^{n} 1_i$ is a sum of n conditionally i.i.d. Bernoulli random variables with the parameter

$1-(1-p)^{m+s}$, under the condition $\sum_{j=1}^{s} T_j = m$. Now, taking $X_n^{(s)} := \sum_{i=1}^{n} 1_i$, we obtain

$$\mathbb{P}\left(X_n^{(s)}=k\right)=\mathbb{P}\left(\sum_{j=1}^{s} T_j<\infty, \sum_{i=1}^{n} 1_i=k\right)=\sum_{m=0}^{\infty} \mathbb{P}\left(\sum_{j=1}^{s} T_j=m, \sum_{i=1}^{n} 1_i=k\right)=$$

Then by expanding the term $\left(1-(1-p)^{m+s}\right)^k$ as $\sum_{l=0}^{k} \binom{k}{l}(-1)^{k-l}(1-p)^{(m+s)l}$ and by grouping terms of binomial series $\sum_{m=0}^{\infty} \binom{m+s-1}{m}((1-q)(1-p)^{n-l})^m$ we get the following:

$$\mathbb{P}\left(X_n^{(s)}=k\right)=q^s\binom{n}{k}\sum_{l=0}^{k}\binom{k}{l}(-1)^l\left(\frac{(1-p)^{n-k+l}}{1-(1-q)(1-p)^{n-k+l}}\right)^s.$$

□

For the calculation of cumulative distribution $C_n(k)=\mathbb{P}(X_n\le k)$, p, q and k should be known. These values can be calculated on the fly, but they can also be calculated in advance and saved on disk. In that case we do not need to repeat calculation for the same k values. Furthermore, since we use a few thousand simulations for each 3-tuple p, q, k, it is easy to see a benefit of the precalculated distributions. Distributions should be precalculated only once and can be used for several networks. The cost of calculation of the distributions for each k up to some k_{max} is proportional to ${k_{max}}^2$. However, the benefit of precalculation is evident in cases when it is necessary to run simulations using different starting parameters [48]. Furthermore, since contact social networks usually have $k_{\max}$ up to tens of thousand, it is necessary to precalculate distribution once for all of them.

A distinction between simulations in the Naive SIR algorithm and the FastSIR algorithm [46] is in the parameter that orders the execution of the simulation. For Naive SIR the simulation is ordered in (discrete) time: the simulation follows the dynamics of infection transfer as it unfolds in time. In the case of FastSIR, the parameter ordering the execution of the simulation is the parameter that we call the generation index. All infected nodes can be classified into generations, according to number of infection transfers from the initially infected node. In particular, the initially infected node has a generation index 0, the nodes that it infects have the generation index 1 and so on. In FastSIR, the simulation starts from the initially infected node (generation 0) and using probability distributions for the number of infected nodes, nodes from generation 1 are determined and the node from generation 0 is recovered. In the n-th step of the simulation, starting from the nodes from generation $n-1$, the nodes from generation n are determined using probability distributions for the number of infected nodes. Then the nodes from the generation $n-1$ are recovered and the simulation proceeds to the next step. Essentially, as a stochastic process, FastSIR captures all infection transfers happening in Naive

Algorithm 2 The FastSIR algorithm

```
**Input:** $(G, \lambda, C)$ where $G$ is contact network and $\lambda$ represents the initial conditions, $C$ is a cumulative distribution for $p$, $q$ and all $k$ values in the network. Initial conditions consist of $p$, $q$, $I$ a queue of initially infected nodes and $S(v)$ is an array indicator of susceptible nodes.
**Output:** array indicator of recovered nodes $R(v)$
while $I$ is not empty do
    dequeue(u, I)
    draw a pseudo random value $r$
    find from $C(k, p, q)$ a maximal value of $k_1$ such that $C(k_1, p, q) \leq r$, where $k_1$ is the number of infected neighbours
    draw from $k$ neighbours $k_1$ nodes $w$
    for each $w$ do
        if $S(w)$ is equal to 1 then
            update $S(w)$ and $R(w)$
            enqueue $(w, I)$
        end if
    end for
end while
output R(v)
```

SIR using different ordering (generation versus time).

Correctness of the FastSIR algorithm

To see correctness of the FastSIR algorithm (see Algorithm 2) we change Naive algorithm in a couple of steps that guarantee equality with respect to all infection transfers happening in Naive SIR process.

- First, all nodes infected directly by initially infected nodes can not recover nor infect their neighbors until the last of the initially infected nodes is recovered. Then, process is repeated so that all infected nodes in moment of recovery of the last initially infected node are defined as the initially infected nodes. It is clear that in this way the probability of infection of any neighbor of initially infected nodes directly by any of initially infected nodes remains unchanged. Since all probabilities of direct transfer of infection remain unchanged until the end of the algorithm, we conclude that this modification leaves a probability of infection of any node unchanged.
- The second step differs from the first step of modifying the Naive SIR algorithm in the way that all nodes except the initially infected node cannot recover nor infect their neighbors until the chosen initially infected node is recovered. Then we choose another node that plays the role of the initially infected node and repeat the process. Probabilities of direct infection of any of the neighbors of initially infected nodes are not changed because the probability of transmission of infection from each initially infected node to any of its neighboring nodes remained the same, and the order in which we have chosen ini-

tially infected nodes does not affect the probability of infection of susceptible neighbors of initially infected nodes.

- The third step is the reduction of all the steps of recovery of chosen initially infected node to a single step; if the initially infected node has m susceptible neighbors, by using a distribution of a random variable of infection we can determine the realization of the number of infected nodes, and then realization of that number of infected nodes among the susceptible m. The probability of direct transmission of infection remains unchanged due to the construction of a random variable of infection.
- Fourth step uses the principle which we prove in the following proposition. It states that in the previous step, number n of adjacent nodes can be taken instead of number m of susceptible nodes, as long as only susceptible adjacent nodes are infected in the process.

Proposition 2.3.2. *Let node v have n neighbors of which $s_1,\dots,s_m$ are susceptible and $i_{m+1},\dots,i_n$ cannot be infected, and let Y be a random variable of number of nodes infected directly by node v. Alternatively, let node v have the same n neighbours $s_1,\dots,s_m,i_{m+1},\dots,i_n$ which are susceptible, and let Z be a random variable of number of nodes infected directly by node v among nodes $s_1,\dots,s_m$. Then Y i Z are identically distributed random variables. Furthermore, in both instances node s_i has the same probability of being directly infected by node v for all i, $1 \leqslant i \leqslant m$.*

Proof. Let node v have m susceptible neighbours and degree n. Probability that k out of m susceptible neighbours end up being infected by node v is obviously $\mathbb{P}(X_m = k)$. Probability that by infecting n neighbours, out of which m can be infected and $n-m$ can not, is obtained as follows:

Probability that i predetermined nodes out of n susceptible nodes become infected is $\mathbb{P}^*(X_n = i)$. We know that only m out of n nodes are actually susceptible, so we have to choose i out of that k nodes which are in the set of m susceptible nodes, and remaining $i-k$ in the set of $n-m$ which can not be infected. That can be done in $\binom{n-m}{i-k}\binom{m}{k}$ different ways. Probability that in the end there are going to be k infected nodes in the set of m susceptible nodes is

$$\sum_{i=0}^{n}\binom{n-m}{i-k}\binom{m}{k}\mathbb{P}^*(X_n = i) = \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\binom{m}{k}\mathbb{P}^*(X_n = i) \tag{2.3}$$

The only thing left to do is compare these two expressions . We will use following relations:

$$\mathbb{P}(X_n = k) = q\binom{n}{k}\sum_{l=0}^{k}\binom{k}{l}(-1)^l\frac{(1-p)^{n-k+l}}{1-(1-q)(1-p)^{n-k+l}} \tag{2.4}$$

$$\mathbb{P}(X_n = k) = \binom{n}{k}\mathbb{P}^*(X_n = k) \tag{2.5}$$

$$\mathbb{P}\left(X_n = k\right) = q\binom{n}{k}\sum_{\mu=0}^{\infty}\left(1-(1-p)^{1+\mu}\right)^{k}\left((1-p)^{1+\mu}\right)^{n-k}(1-q)^{\mu} \tag{2.6}$$

We have

$$\mathbb{P}^{*}(X_m = k) \overset{2.5,2.6}{=} q\sum_{\mu=0}^{\infty}(1-(1-p)^{1+\mu})^{k}((1-p)^{1+\mu})^{m-k}(1-q)^{\mu} =$$

$$= q\sum_{\mu=0}^{\infty}(1-(1-p)^{1+\mu})^{k}((1-p)^{1+\mu})^{m-k}(1-q)^{\mu} *$$

$$\underbrace{*\sum_{i=0}^{n-m}\binom{n-m}{i}(1-(1-p)^{1+\mu})^{i}((1-p)^{1+\mu})^{n-m-i} =}_{=1}$$

$$= q\sum_{\mu=0}^{\infty}\left(1-(1-p)^{1+\mu}\right)^{k}\left((1-p)^{1+\mu}\right)^{m-k}(1-q)^{\mu} *$$

$$* \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\left(1-(1-p)^{1+\mu}\right)^{i-k}\left((1-p)^{1+\mu}\right)^{n-m-i+k} =$$

$$= q\sum_{\mu=0}^{\infty}(1-q)^{\mu}\sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\left(1-(1-p)^{1+\mu}\right)^{i}\left((1-p)^{1+\mu}\right)^{n-i} =$$

$$= \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\underbrace{\sum_{\mu=0}^{\infty}q\left(1-(1-p)^{1+\mu}\right)^{i}\left((1-p)^{1+\mu}\right)^{n-i}(1-q)^{\mu}}_{=\mathbb{P}^{*}(X_n=i)} = \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\mathbb{P}^{*}\left(X_n = i\right)$$

which implies

$$\mathbb{P}^{*}\left(X_m = k\right) = \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\mathbb{P}^{*}\left(X_n = i\right) \tag{2.7}$$

and obviously (2.7) ⇒ (2.8)

$$\mathbb{P}\left(X_m = k\right) = \sum_{i=k}^{n-m+k}\binom{n-m}{i-k}\binom{m}{k}\mathbb{P}^{*}\left(X_n = i\right) \tag{2.8}$$

By taking $m = 1$ in both instances in equations (2.3) and (2.8) we obtain the same probability of node s_i being directly infected by node v for all i, $1 \leqslant i \leqslant m$. □

Alternative proof of the correctness is mapping the FastSIR algorithm to the single realization of the semi-directed epidemic percolation networks due to their isomorphism with time-homogeneous SIR model [19]. Starting from the contact network the semidirected epidemic percolation network under the discrete time SIR model is generated by: (i) choosing the recovery time τ_i for each node in a network independently from a geometric distribution, (ii) for each pair of connected nodes (i, j) converting the undirected edge to a directed from i to j with the probability $(1-(1-p)^{\tau_i})(1-p)^{\tau_j}$, to a directed from j to i with the probability $(1-(1-p)^{\tau_j})(1-p)^{\tau_i}$ and destroying the edge with the probability $(1-p)^{\tau_i}(1-p)^{\tau_j}$. The final outcome of a single realization of the SIR process from an initially infected node i is equal to the out-component of node i in the epidemic percolation network [19], which is proved by showing that the first infection time is finite if and only if the node in the out-component of node i. Therefore, the FastSIR algorithm can be seen as a fast algorithm for drawing the directed edges only in the out-component of a initially infected node in the epidemic percolation network. The transmission probabilities: $\{(1-(1-p)^{\tau_i})(1-p)^{\tau_j},(1-(1-p)^{\tau_j})(1-p)^{\tau_i},(1-p)^{\tau_i}(1-p)^{\tau_j}\}$ are encoded in the probability distribution $\mathbb{P}^*(X_n = k)$ as the recovery τ_i is a sample from a geometric distribution which is a instance of a negative binomial distribution, which is used in deriving the probability infectious distribution $\mathbb{P}^*(X_n = k)$.

Time complexity analysis of the FastSIR algorithm

Here, we examine the average case running time and space complexity of the FastSIR algorithm by using the big-O notation (asymptotic upper bound within a constant factor) [47].

Proposition 2.3.3. *The average case running time of the FastSIR algorithm $\overline{T_f}$ on an arbitrary network structure is equal to:*

$$\overline{T_f} = O(\mathbb{E}[X]\,\overline{k}),$$

where $\mathbb{E}[X]$ denotes the total expected number of infected nodes and $\overline{k}$ average degree in the network.

Proof. Let us start with one infected node and its k (degree) susceptible neighbours. Since the distribution of the number of infected neighbours is precalculated and it is possible to access the data with $O(1)$, we can neglect that to the overall cost. The first step is uniformly choosing a value for the cumulative distribution. Since the parameters p, q and k are known, we should find the appropriate number of infected nodes k_1 for that realisation. From the fact that there are $k+1$ possible values we can find k_1 in $log(k)$ steps using the binary search algorithm. In the next step, a random sample of k_1 nodes should be chosen, that would be infected, from k of them. For that operation, the calculation cost is proportional to $min(k_1, k-k_1)$ [49]. In the last step, the infection should be transmitted to k_1 neighbouring nodes, so the calculation cost is proportional to k_1. The overall running time for one infected node T_f^1 and k susceptible neighbours can be calculated from the sum of costs for all three steps and it is equal to

$$T_f^1 = c_1 log(k) + c_2 min(k_1, k - k_1) + c_3 k_1 = O(k), \tag{2.9}$$

where c_1, c_2 and c_3 are constants. Since $k_1 < k$, the average case running time for one node is $O(k)$. Hence, it does not depend on $1/q$. The total average case running time $\overline{T_f}$ is the sum of average times $\overline{T_f^i}$ for all infected nodes v_i because the main while loop of the FastSIR algorithm (see Algorithm 10) executes sequentially. This sum $\overline{T_f^1} + \overline{T_f^2} + ... + \overline{T_f^n}$ has $\mathbb{E}[X]$ terms which have $O(k)$ average case running time. Therefore, the average case running time $\overline{T_f}$ is equal to O ($\mathbb{E}[X]\overline{k}$) . $\square$

Exact running times of the FastSIR algorithm (non-asymptotic regime) can be influenced by the network structure and the value of parameter p. Because of this in some special cases for a smaller part of (p,q) space the FastSIR algorithm could be slightly slower. The experiments measuring the FastSIR running time can be found in section 2.4.

The space complexity of the FastSIR algorithm

The space complexity S of the FastSIR algorithm with respect to the number of links L, the number of nodes N and the sum of all distinct degrees in network K is equal to:

$$S[L,N,K] \approx \underbrace{2L}_{G} + \underbrace{K}_{C} + \underbrace{N}_{I} + \underbrace{N}_{S(v)} + \underbrace{N}_{R(v)} = 2L + K + 3N,$$

where the first term denotes the space complexity of contact network G (adjacency list), the second term denotes the space complexity of cumulative distributions C for all distinct degrees k_i in G, the third term denotes space complexity of a queue of infected nodes I, the next term denotes the space complexity of the vector indicator of susceptible nodes $S(v)$ and the last term denotes the space complexity of the vector indicator of recovered nodes $R(v)$. Note that the $S(v)$ and $R(v)$ can be implemented as a bitset structure to further reduce memory consumption.

In connected networks $L \geqslant N$ and $2L \geqslant K$ and then the space complexity S of the FastSIR algorithm is:

$$S[L,N,K] = O(L).$$

The implementation of distribution precalculation

Looking at the cumulative distribution formula, it can be seen that a calculation cost is proportional to ${k_{max}}^4$. A speed up can be achieved using the fact that binomial coefficient values and the fraction part of the formula are used repeatedly, so by caching them we can obtain lower calculation costs. However, we achieved a further speed up using the recursive formula (2.10).

Proposition 2.3.4. *For each $k \neq 0$*

$$\mathbb{P}(X_n = k) = \frac{n}{k}\mathbb{P}(X_{n-1} = k-1) - \frac{n-k+1}{k}\mathbb{P}(X_n = k-1) \tag{2.10}$$

Proof 1.

$$\mathbb{P}\left(X_n = k\right) = q\binom{n}{k}\sum_{l=0}^{k}\binom{k}{l}\frac{(-1)^l(1-p)^{n-k+l}}{1-(1-q)\left(1-p\right)^{n-k+l}}$$

$$= q\binom{n}{k}\sum_{l=0}^{k}\left[\binom{k-1}{l-1}+\binom{k-1}{l}\right]\frac{(-1)^l(1-p)^{n-k+l}}{1-(1-q)\left(1-p\right)^{n-k+l}}$$

$$= \underbrace{q\binom{n}{k}\sum_{l=0}^{k}\binom{k-1}{l-1}\frac{(-1)^l(1-p)^{n-k+l}}{1-(1-q)\left(1-p\right)^{n-k+l}}}_{=:S_1} + \underbrace{q\binom{n}{k}\sum_{l=0}^{k}\binom{k-1}{l}\frac{(-1)^l(1-p)^{n-k+l}}{1-(1-q)\left(1-p\right)^{n-k+l}}}_{=:S_2}$$

$$S_1 = -\frac{n-k+1}{k}q\binom{n}{k-1}\sum_{l=0}^{k-1}\binom{k-1}{l}\frac{(-1)^l(1-p)^{n-(k-1)+l}}{1-(1-q)\left(1-p\right)^{n-(k-1)+l}} =$$

$$= -\frac{n-k+1}{k}\mathbb{P}\left(X_n = k-1\right)$$

$$S_2 = q\binom{n}{k}\sum_{l=0}^{k-1}\binom{k-1}{l}\frac{(-1)^l(1-p)^{n-k+l}}{1-(1-q)\left(1-p\right)^{n-k+l}}$$

$$= \frac{n}{k}q\binom{n-1}{k-1}\sum_{l=0}^{k-1}\binom{k-1}{l}\frac{(-1)^l(1-p)^{(n-1)-(k-1)+l}}{1-(1-q)\left(1-p\right)^{(n-1)-(k-1)+l}}$$

$$= \frac{n}{k}\mathbb{P}\left(X_{n-1} = k-1\right) \Rightarrow \mathbb{P}\left(X_n = k\right) = \frac{n}{k}\mathbb{P}\left(X_{n-1} = k-1\right) - \frac{n-k+1}{k}\mathbb{P}\left(X_n = k-1\right)$$

□

This was an algebraic proof, but we can also make a different proof by starting from one probabilistic rule.

Proof 2. Let us denote the probability that specific k out of n nodes is infected with $\mathbb{P}^*\left(X_n = k\right)$. The relation with the $\mathbb{P}\left(X_n = k\right)$ is the following:

$$\mathbb{P}\left(X_n = k\right) = \binom{n}{k}\mathbb{P}^*\left(X_n = k\right),$$

as we have fixed k nodes. If we observe the event that the node with $n-1$ edges infected $k-1$ specific nodes and if we add one more edge to that node the disease can either be transmitted through it or not, which are disjunct events and therefore:

$$\mathbb{P}^*\left(X_{n-1}=k-1\right)=\mathbb{P}^*\left(X_n=k\right)+\mathbb{P}^*\left(X_n=k-1\right).$$

Now if we multiply this with relation with $\binom{n}{k}$ we get:

$$\binom{n}{k}\mathbb{P}^*\left(X_{n-1}=k-1\right)=\mathbb{P}\left(X_n=k\right)+\binom{n}{k}\mathbb{P}^*\left(X_n=k-1\right)$$

$$\frac{\binom{n}{k}}{\binom{n-1}{k-1}}\mathbb{P}\left(X_{n-1}=k-1\right)=\mathbb{P}\left(X_n=k\right)+\frac{\binom{n}{k}}{\binom{n}{k-1}}\mathbb{P}\left(X_n=k-1\right)$$

$$\implies \mathbb{P}\left(X_n=k\right)=\frac{n}{k}\mathbb{P}\left(X_{n-1}=k-1\right)-\frac{n-k+1}{k}\mathbb{P}\left(X_n=k-1\right).$$

□

By using this recursive formula, the computation cost is proportional to ${k_{max}}^2$. It is very important to mention that in the programming of the cumulative distribution one should be very careful with precision. Because of that, we use a multiple precision library for this calculation. Empirically, we obtained that it is safe to set the precision to be at least 0.8 times degree bits. The minimum precision is 64 bits. During the testing of the calculation time, we noticed that the cost for large degree values predominantly depended on the precision used. The cumulative distribution values should be precalculated for a specific maximum degree only once and they can be used for all networks that have degrees less than the maximum one. We consider that 50000 is a high enough value of a degree for the majority of networks. A similar recursive formula can be used when the random variable of time of recovery for each node is distributed as the negative binomial probability distribution.

Parallelization of the algorithm

As in similar algorithms [30], parallelization can be performed by a partition of networks using MPI. Since we used a large number of repetitions, it can also be naively parallelized by performing each repetition on a separate core. A parallelization using GPUs is also possible [50]. In this thesis, we have only used parallelization by MPI standard.

2.4 The FastSIR experiments

In this section, we describe detailed performance profiling and the analysis of implementations of the FastSIR algorithm and the Naive SIR algorithm on our test server. The server has 4 Quad Core 2.4 GHz Intel E5330 processors and 50 GB of RAM memory. For test purposes, we use only one core for each test. Algorithms are implemented in C using the igraph [51] and the gmp libraries [52].

The analysis was performed on several empirical networks: the network of 2003 condensed matter collaborations (cond-mat 2003) introduced in [53], an undirected, unweighted network representing the topology of the US Western States Power Grid (power grid) [11], the network of co-authorships between scientists posting preprints on the Astrophysics E-Print Archive between January 1, 1995 and December 31, 1999 (astrophysics) [53], a symmetrized snapshot of the structure of the Internet at the level of autonomous systems, reconstructed from BGP tables posted by the University of Oregon Route Views Project (Internet) [54] and the network of Live Journal users (Live Journal) [55]. Table 2.1 shows the basic information for networks mentioned above.

Table 2.1: Basic network parameters

Network	no of nodes	no of links	k_{max}	$\overline{k}$	sum of distinct degrees
Power grid	4 941	6 594	19	2.7	142
Cond-mat 2003	27 519	116 181	202	8.4	8 619
Astro physics	14 845	119 652	360	16.1	16 737
Internet	22 963	48 436	2390	4.2	32 118
Live Journal	5 189 809	77 365 447	15 023	29.6	2 503 563

For each analysis, we measured the running time of the Naive SIR algorithm and the FastSIR algorithm excluding the time needed for loading network structure from the disk. Loading network structure data (adjacency list) from a disk were not measured in the running time analysis of both algorithms. However, loading precalculated probability distributions from a disk was measured in the running time analysis of the FastSIR algorithm. Also, we measured the execution time for distribution precalculation. We studied the entire (p,q) parametric space of the SIR model: a $[0,1]\times[0,1]$ square. The step value for both p and q was 0.1. Each simulation was started from the same node for each algorithm, and it was performed 2000 times. The upper bound for a memory consumption for all experiments was 9 GB. Although some authors use only several dozen of repetitions, we consider that is not sufficient to obtain stable results in the bimodal part of the phase space. Note, that the variables measuring the extent of disease

Table 2.2: Running time in seconds for 2000 simulations, p = 0.2, 0.5, 0.8 and q = 0.1

Network	p=0.2		p=0.5		p=0.8	
	Naive SIR	FastSIR	Naive SIR	FastSIR	Naive SIR	FastSIR
Power grid	3.2	0.4	7.0	0.9	7.1	0.9
Cond-mat 2003	67.7	9.6	63.3	8.6	61.2	7.9
Astro Physics	44.1	7.0	41.2	5.9	39.9	5.1
Internet	42.5	5.0	41.6	4.9	40.5	4.7
Live Journal	50 683	6 699	48 373	5 635	47 531	5 078

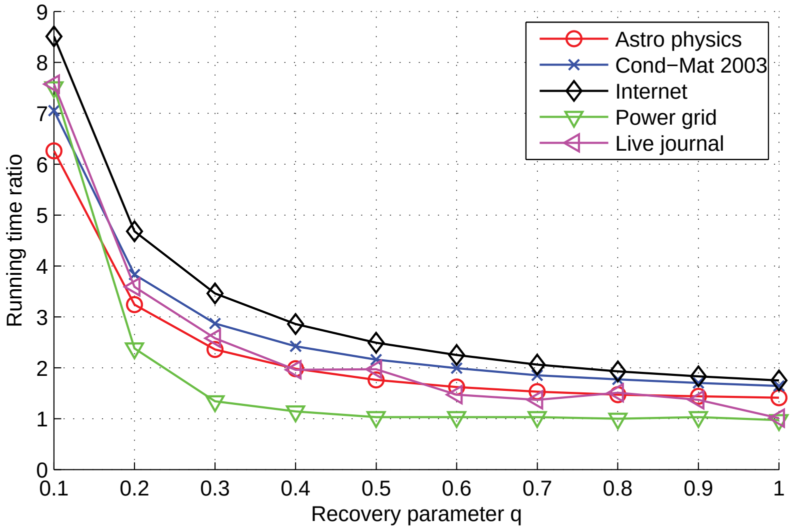


Figure 2.2: Running time ratios for the Naive SIR algorithm and the FastSIR algorithm. Parameters: 2000 simulations, p = 0.2, q = 0.1 to 1.

spreading are in general characterized by a bimodal probability distribution [23]. The results of running time for p values of 0.2, 0.5 and 0.8 and q value of 0.1 for all tested networks are presented in Table 2.2. In addition, Table 2.3 presents the results for p values of 0.2, 0.5 and 0.8 and q values between 0.1 and 1. Graphs of the results obtained for p values of 0.2, 0.5 and 0.8 and different values of q for all networks are presented in Figure 2.2, Figure 2.3 and Figure 2.4, respectively. Those figures show the ratio of running time between Naive SIR and FastSIR.

Results differ between networks, but the trend of the ratio is approximately proportional to $1/q$. When the value of q is near one, the running time ratio differs depending on the network and the value of p. It can be seen that the results are in accordance with the analysis above. When p is small ($p = 0.2$), the FastSIR algorithm is faster or equal to Naive SIR for all q values. But, when p has the value of 0.5, the Naive SIR algorithm is faster for larger q values

Table 2.3: Running times in seconds for Live Journal network. Parameters: 2000 simulations, p = 0.2, 0.5 and 0.8, q = 0.1 to 1

q	p=0.2		p=0.5		p=0.8	
	Naive SIR	fastSIR	Naive SIR	fastSIR	Naive SIR	fastSIR
0.1	50 683	6 699	48 373	5 635	47 531	5 078
0.2	25 841	7 200	24 398	6 314	24 067	5 357
0.3	18 550	7 200	16 580	6 841	16 276	5 609
0.4	13 686	6 987	12 870	7 259	12 329	5 843
0.5	13 197	6 704	10 394	7 591	9 951	6 060
0.6	9 394	6 400	8 720	7 859	8 345	6 253
0.7	8 301	6 073	7 513	8 072	7 185	6 429
0.8	8 744	5 805	6 622	8 250	6 293	6 592
0.9	7 521	5 508	5 869	8 555	5 597	6 749
1	5 291	5 259	5 064	8 666	5 082	7 777

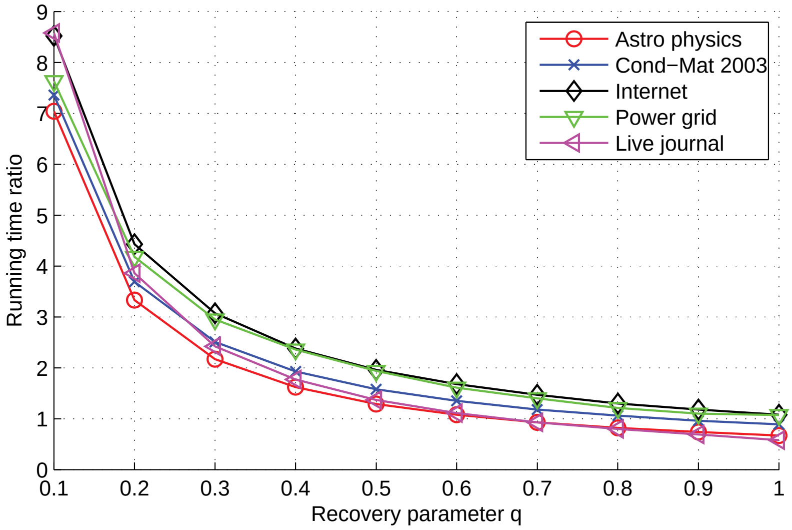


Figure 2.3: Running time ratios for the Naive SIR algorithm and the FastSIR algorithm. Parameters: 2000 simulations, p = 0.5, q = 0.1 to 1.

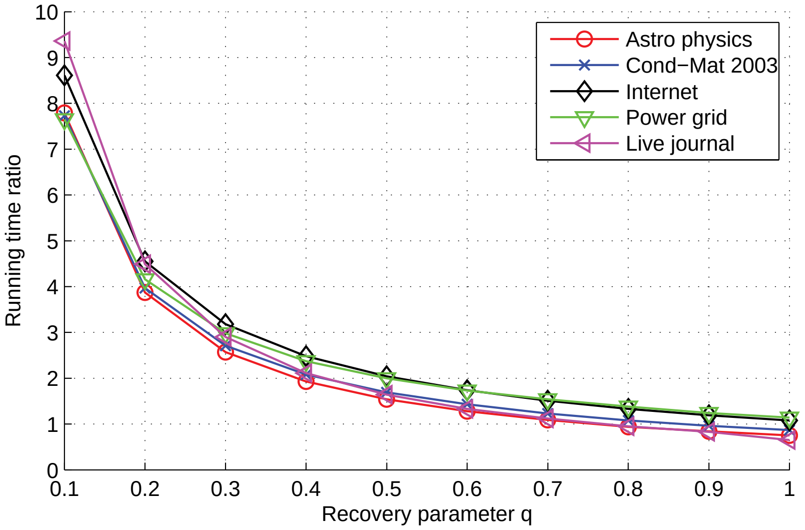


Figure 2.4: Running time ratios for the Naive SIR algorithm and the FastSIR algorithm. Parameters: 2000 simulations, p = 0.8, q = 0.1 to 1.

for almost all networks. In addition, when p has the value 0.8, the Naive SIR algorithm can be faster for some networks and q values very close to one. It is obvious that the FastSIR algorithm is significantly faster only for q values less than 0.5 where the speedup is greater than 2. Thus, e.g. for q value of 0.1, the ratio is between 7 and 9.5 depending on the network and the value of p. It is important to note that the simulation time of the Naive SIR algorithm is most critical just for small q values.

It is very important to emphasize that the results for the Live Journal network of 5 million nodes and 77 million links are very fast. The average case running time for one simulation for p of 0.2 was between 2 and 4 seconds for the FastSIR algorithm. Furthermore, it should be stressed that the results were achieved without parallelization. Hence, the parallel implementations of both algorithms can be used for very large networks.

2.5 Continuous FastSIR algorithm

Note, that the FastSIR algorithm runs in generation times and that we can change the model of spreading the disease just by changing the infection distribution of first neighbourhood. One possible extension is the continuous time SIR model, where infected node transmits the disease to susceptible node at an average rate β and infected nodes recover at the constant rate γ. So the probability of recovering in any short time interval Δt is $\gamma\Delta t$ and the probability of transmitting the disease in any short time interval Δt is $\beta\Delta t$.

Proposition 2.5.1. *The probability density function of random variable X_n, which measures that the infected node infects x neighbours out of total n susceptible neighbours in the limit of the* *continuous time under the SIR model (β, γ) is:*

$$f_{X_n}(x) = \gamma \sum_{k=0}^{n} \binom{n}{k} \delta(x-k) \frac{\Gamma(k+1)\Gamma(\frac{\gamma+\beta(n-k)}{\beta})}{\beta\Gamma(1+k+\frac{\gamma+\beta(n-k)}{\beta})}, \tag{2.11}$$

where $\Gamma(x)$ is the gamma function.

Proof. The probability of recovering in any short time interval Δt is $\gamma\Delta t$ and the probability that the node is still infected after a τ time is:

$$\lim_{\Delta t \to 0} (1-\gamma\Delta t)^{\frac{\tau}{\Delta t}} = \lim_{\frac{\tau}{\Delta t} \to \infty} (1 + \frac{-\gamma\tau}{\frac{\tau}{\Delta t}})^{\frac{\tau}{\Delta t}} = e^{-\gamma\tau}, \tag{2.12}$$

and the probability that the node remains infected this long and then recovers in the interval $\tau +$ dτ is: $e^{-\gamma\tau}\gamma \mathrm{d}\tau$, which is a standard exponential distribution with parameter γ. This is a standard formulation of the SIR continuous model [56]. Likewise, the probability that transmission does not happen if the infected node remains infected for τ time long is $e^{-\beta\tau}$.

Now, we will find the corresponding $f(x)$ probability density function of X_n when the time passes continuously. First, we find the conditional pdf $f_{X_n|T}$, where $T = t$ is the time of recovery of infected node. We obtain the conditional pdf $f_{X_n|T}$ by using the Binomial distribution $B(n, 1-e^{-\beta t})$ with the exponential distribution $\varepsilon(\beta)$ for transmission time event and Dirac delta distributions $\delta(x)$:

$$f_{X_n|T} = \sum_{k=0}^{n} \binom{n}{k} e^{(-\beta t)(n-k)} (1-e^{-\beta t})^k \delta(x-k). \tag{2.13}$$

Now, we obtain the joint pdf $f_{X_n,T} = f_{X_n|T} f_T$, where the f_T is the exponential distribution for recovery $\varepsilon(\gamma)$:

$$f_{X_n,T} = \gamma e^{-\gamma t} \sum_{k=0}^{n} \binom{n}{k} e^{(-\beta t)(n-k)} (1-e^{-\beta t})^k \delta(x-k). \tag{2.14}$$

And finally, we obtain the marginal pdf $f_{X_n} = \int_0^{+\infty} f_{X_n|T} f_T \mathrm{d}t$:

$$f_{X_n} = \gamma \sum_{k=0}^{n} \delta(x-k) \binom{n}{k} \int_0^{+\infty} e^{-\gamma t} e^{(-\beta t)(n-k)} (1-e^{-\beta t})^k \mathrm{d}t. \tag{2.15}$$

After integration we get the following expression:

$$f_{X_n}(x) = \gamma \sum_{k=0}^{n} \binom{n}{k} \delta(x-k) \frac{\Gamma(k+1)\Gamma(\frac{\gamma+\beta(n-k)}{\beta})}{\beta\Gamma(1+k+\frac{\gamma+\beta(n-k)}{\beta})}. \tag{2.16}$$

□

By using the cdf of this function, we can run the FastSIR algorithm also in continuous time. The $f_{X_n}(x)$ is a valid pdf: (1) $\int_0^{+\infty} f_{X_n}(x)\mathrm{d}x = 1$ and (2) $f_{X_n}(x) \geq 0$ (all arguments to the gamma functions are real and non-negative).

A similar recursive relation holds for continuous distribution.

Proposition 2.5.2. *For each $k \neq 0$:*

$$f_{X_n}(k) = \frac{n}{k} f_{X_{n-1}}(k-1) - \frac{n-k+1}{k} f_{X_n}(k-1) \tag{2.17}$$

The proof is equivalent to the proof 2 from the proposition 2.10 where $\mathbb{P}(X_n = k)$ denotes value $f_{X_n}(k)$.

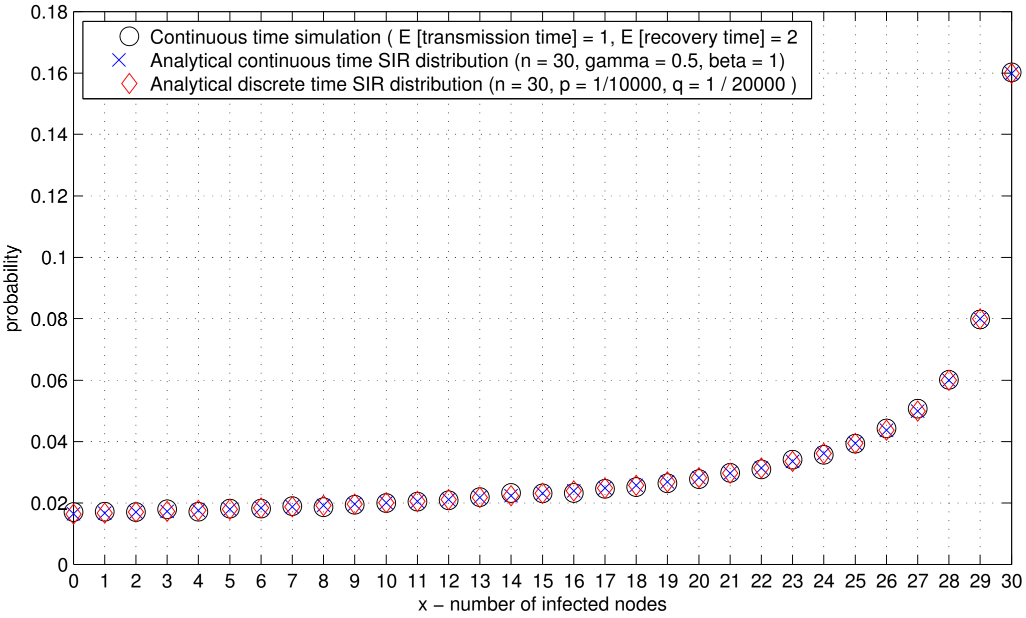


Figure 2.5: The correspondence of the continuous time Monte Carlo simulations on a bipartite graph, continuous time SIR distribution and discrete time SIR distribution. The black circles indicate the distribution of infected nodes of 10^6 continuous time Monte Carlo simulations on a bipartite graph with 1 infected node in the first layer and 30 susceptible nodes in the second layer at the initial moment, where the expected transmission time equals to 1 and expected recovery time equals to 2. The blue crosses indicate analytical continuous time SIR distribution with parameters: $n = 30$, $\gamma = 0.5$ and $\beta = 1$. The red diamonds indicate analytical discrete time SIR distribution with parameters $n = 30$, $p = 0.0001$, $q = 0.00005$.

The discrete time process can approximate the continuous process if the frequency of discrete sampling $f = \frac{1}{\Delta t}$ is high or the granularity Δt of discrete step simulation is arbitrary low w.r.t. continuous time. The connection between expected time of recovery of a node in continuous and discrete time is $E_q[\Delta t] = E_\gamma[t] * f$. Note, that here the term $E_q[\Delta t]$ denotes the expected number of discrete steps for a recovery event and $E_\gamma[t]$ is the expected time for a recovery event in continuous time. This approximation can hold only if the expected time of recovery and transmission events are a few orders of magnitude higher than the granularity of discrete simulation Δt. For example, if we set the expected time of recovery in continuous time to be 2 [days] and the sampling frequency $f = 10000$, then the expected number of discrete steps $[1day/f] \approx [8.6min]$ should be $E_{\gamma}[t] * f = 20000$ for a recovery event to occur. The same holds for expectation of p and β. In the Figure 2.5, we give the continuous version and discrete version of X_n probability for the corresponding parameters with $\Delta t = 10000$, calculated for $n = 30$ and the Monte Carlo simulation, where transmission and recovery times are distributed with exponential distribution according to the previous example.

Chapter 3

Backward in time - single source epidemic detection

3.1 Problem formulation

In this section we formulate the problem of the source detection in a network. Let us define the random vector $\vec{R} = (R(1), R(2), ..., R(N))$ that indicates which nodes got infected up to a predefined temporal threshold T (a random variable or a constant). The random variable $R(i)$ is a Bernoulli random variable, which has the value of 1 if the node i got infected before time T from the start of the epidemic process and the value of 0 otherwise. Let us assume that we have observed one spatio-temporal epidemic propagation realization $\vec{r}_*$ of $\vec{R}$ of the SIR process defined by (p, q, T) over a network G, and we want to infer which nodes from the set S are the most likely to be the source of realization $\vec{r}_*$ for the SIR process (p, q, T) on G. $S = \{\theta_1, \theta_2, ..., \theta_m\}$ is the finite set of possible source nodes that is defined by observed infected or recovered set of nodes prior to the moment T in the network.

The node with the highest posterior probability for being the source of the epidemic spread for a given realization $\vec{r}_*$ is $\hat{\Theta}_{MAP} = \arg\max_{\theta_i \in S} P(\Theta = \theta_i | \vec{R} = \vec{r}_*)$. By applying the Bayes theorem, we get the following expression:

$$P(\Theta = \theta_i | \vec{R} = \vec{r}_*) = \frac{P(\vec{R} = \vec{r}_* | \Theta = \theta_i) P(\Theta = \theta_i)}{\sum\limits_{\theta_k \in S} P(\vec{R} = \vec{r}_* | \Theta = \theta_k) P(\Theta = \theta_k)}. \tag{3.1}$$

Unless stated otherwise, due to the simplicity of the notation in the rest of the thesis, we do not put the following variables: p, q, T, G to the condition $P(\Theta = \theta_i | \vec{R} = \vec{r}_*, T, p, q, G)$. Thus, the core of the source detection problem is the determination of the source probability distribution over nodes that have been infected/recovered in a given observed epidemic realization. For simplicity, we will assume that all nodes have the same prior probability and therefore the maximum posterior probability node is equal to the maximum likelihood node (ML node), but the methodology is also applicable in cases when the source prior probabilities are not uniform. In Figure 3.1, we show the visualization of the solution of the source detection problem.

Now, we will state the assumptions that we use to solve the formulated problem:

- complete knowledge about network: $G = (V, E)$,
- complete knowledge about contagion process: SIR (p, q),
- complete knowledge about temporal parameter: T and
- complete knowledge of nodes which were infected prior to T: $\vec{r}_*$.

Later, we will demonstrate that our inference framework is applicable even when the complete knowledge about the contagion process (p, q, T), the network structure G or the node states $\vec{r}_*$) are relaxed.

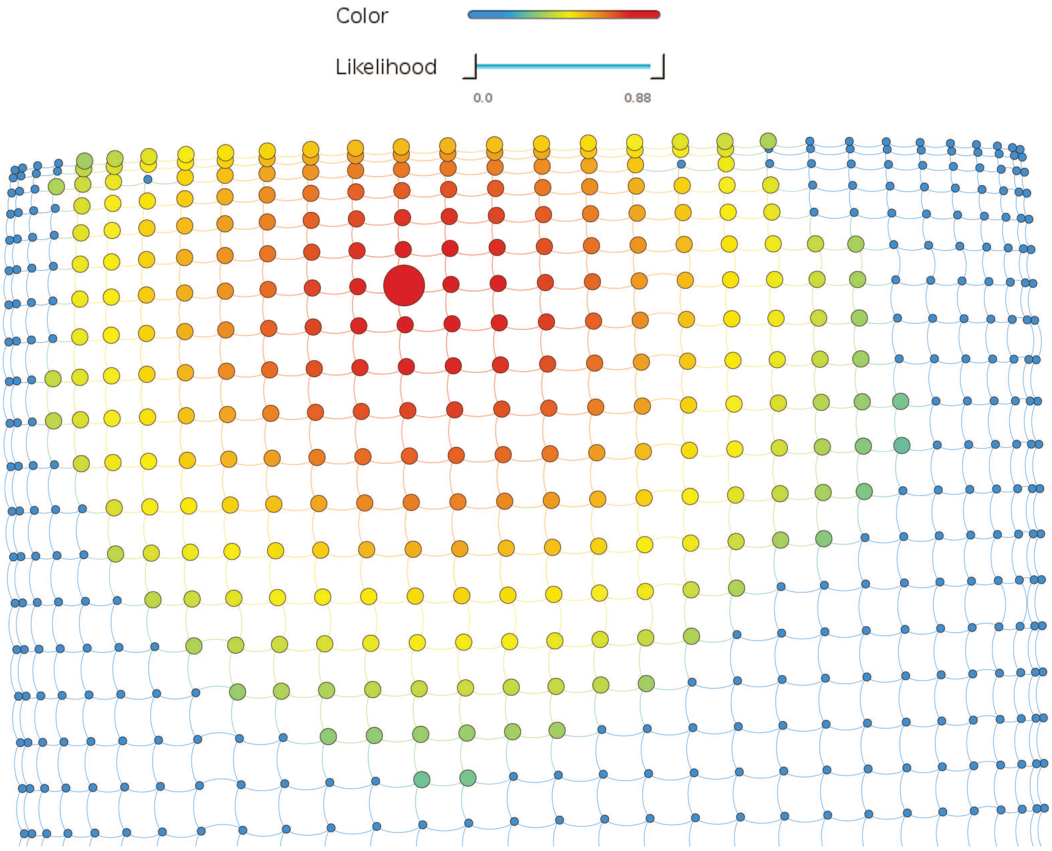


Figure 3.1: Colour visualization of the source likelihood estimates on a regular grid 30x30 (4-connected neighbourhood) for $p = 0.6$, $q = 0.3$, $T = 15$. Nodes which are not in the epidemic realization are represented with the smallest size circles while the infected nodes have bigger size circles and the biggest node is the origin of the epidemic realization. The blue colour represents the smallest likelihood estimate, the yellow colour represents the middle range values while the red colour represents the highest likelihood estimates.

3.2 The AUCDF and the AvgTopk – heuristic similarity-based methods

In this section, we define the two similarity-based methods for source detection. Let us define the function $\varphi(\vec{r}_1, \vec{r}_2)$, which measures the similarity between two epidemic realizations or subgraphs of the underlying network: $\vec{r}_1$ and $\vec{r}_2$. Then a random variable $\varphi(\vec{r}_*, \vec{R}_\theta)$ measures the similarity φ between a fixed realization $\vec{r}_*$ and a random realization that comes from SIR process with the source being θ. Note, that the random variables are denoted with the capital letter $\vec{R}_\theta$ and the particular instance or value of the corresponding random variable is denoted with the small letter $\vec{r}_\theta$. The empirical distribution function is the unbiased estimator of the following cumulative distribution function:

$$\hat{F}(x) = \hat{P}(\varphi(\vec{r}_*, \vec{R}_\theta) \leq x) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[0,x\rangle}\left(\varphi(\vec{r}_*, \vec{R}_{\theta,i})\right)}{n}, \tag{3.2}$$

where $\mathbf{1}_{[0,x\rangle}$ is a characteristic function defined as:

$$\mathbf{1}_{[0,x\rangle}(y) = \begin{cases} 1 & : y \in [0,x\rangle, \\ 0 & : else. \end{cases} \tag{3.3}$$

Then, its probability density function is calculated like this:

$$PDF(x) = \frac{d}{dx}\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} \delta\left(x - \varphi(\vec{r}_*, \vec{R}_{\theta,i})\right), \tag{3.4}$$

where $\delta(x)$ is the Dirac delta distribution.
Central limit theorem states that pointwise, $\hat{F}(x) - F(x)$ has asymptotically normal distribution. The rate at which this convergence happens is bounded by Berry–Esseen theorem. This implies that the rate of convergence is bounded by $O(1/\sqrt{n})$, where n is the number of simulations.

Next, we define two measures ($XNOR$ and Jaccard) that are used to determine the similarity φ. The first one is a binary NOT XOR function or $XNOR(\vec{r}_1, \vec{r}_2)$ that counts the number of the corresponding non-infected and infected nodes in realizations $\vec{r}_1$ and $\vec{r}_2$:

$$XNOR(\vec{r}_1, \vec{r}_2) = \frac{1}{N}\sum_{k \in V} \psi_{\oplus}(\vec{r}_1(k), \vec{r}_2(k)), \tag{3.5}$$

where N is the total number of nodes and $\psi_{\oplus}(m_1,m_2)$ function is defined as:

$$\psi_{\oplus}(m_1,m_2) = \begin{cases} 1 & : (m_1 = 1 \text{ and } m_2 = 1) \text{ or } (m_1 = 0 \text{ and } m_2 = 0)\ , \\ 0 & : \text{else.} \end{cases} \tag{3.6}$$

In other words, $\psi(m_1,m_2)$ is equal to one only if two nodes were infected or they did not get infected prior to temporal threshold T.

The second similarity measure is the well known Jaccard measure, which in our case counts the number of corresponding infected nodes in $\vec{r}_1$ and in $\vec{r}_2$ normalized by the number of corresponding infected nodes in $\vec{r}_1$ or in $\vec{r}_2$.

$$Jaccard(\vec{r}_1,\vec{r}_2) = \frac{|\vec{r}_1 \wedge \vec{r}_2|}{|\vec{r}_1 \vee \vec{r}_2|} = \frac{\sum_{k\in V} \psi_{\wedge}(\vec{r}_1(k),\vec{r}_2(k))}{\sum_{k\in V} \psi_{\vee}(\vec{r}_1(k),\vec{r}_2(k))}, \tag{3.7}$$

where $\psi_{\wedge}(m_1,m_2)$ is binary AND function and $\psi_{\vee}(m_1,m_2)$ is binary OR function.

In the following text the $\varphi_x(\vec{r}_1,\vec{r}_2)$ will denote the similarity calculated with $\overline{XNOR}(\vec{r}_1,\vec{r}_2)$ function and $\varphi_J(\vec{r}_1,\vec{r}_2)$ will denote the similarity calculated with $Jaccard(\vec{r}_1,\vec{r}_2)$ function. In order to speed the similarity matching between realizations, we use the bitwise operations (XOR, NOT, AND) and bit count with Biran-Kernignan method [57].

Now, we define two variants of likelihood estimation functions: AUCDF and AvgTopK.

Algorithm 3 AUCDF estimation function $(G,p,q,\vec{r}_*,T,\theta,n)$

```
**Input:** $G$ - contact network, $(p,q)$ - SIR process parameters, $\vec{r}_*$ - observed realization prior to some temporal threshold $T$, $\theta$ - source for which likelihood is calculated, $n$ - a number of simulations
**for** $i$ = 1 to $n$ (number of simulations) **do**
    - Run SIR simulation $(p,q)$ with $\Theta = \theta$ and obtain epidemic realization $\vec{R}_{\theta,i}$, ending at the temporal threshold $T$;
    - Calculate and save $\varphi(\vec{r}_*,\vec{R}_{\theta,i})$ ;
**end for**
- Calculate empirical distribution function:
    $\hat{P}(\varphi(\vec{r}_*,\vec{R}_\theta) \leq x) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[0,x\rangle}(\varphi(\vec{r}_*,\vec{R}_{\theta,i}))}{n}$
- Estimate the likelihood using the area under the empirical cumulative distribution:
    $AUCDF_\theta = \int_0^1 \hat{P}(\varphi(\vec{r}_*,\vec{R}_\theta) \leq x)\mathrm{d}x$
**Output:** $\hat{P}(\vec{R} = \vec{r}_*|\Theta = \theta) = 1 - AUCDF_\theta$ likelihood for $\theta$;
```

Algorithm 4 AvgTopK likelihood estimation function $(G, p, q, \vec{r}_*, T, \theta, n)$

```
**Input:** $G$ - contact network, $(p,q)$ - SIR process parameters, $\vec{r}_*$ - observed realization prior to some temporal threshold $T$, $\theta$ - source for which likelihood is calculated, $n$ - a number of simulations
for $i$ = 1 to $n$ (number of simulations) do
    - Run SIR simulation $(p,q)$ with $\Theta = \theta$ and obtain epidemic realization $\vec{R}_{\theta,i}$, ending at the temporal threshold $T$;
    - Calculate and save $\varphi(\vec{r}_*, \vec{R}_{\theta,i})$ ;
end for
- Sort the scores $\left\{\varphi(\vec{r}_*, \vec{R}_{\theta,i})\right\}$ in descending order;
- Average top $k$ highest scores:
    $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta) = \frac{1}{k} \sum_{i=1}^{k} \left\{\varphi(\vec{r}_*, \vec{R}_{\theta,i}))\right\}_{sorted}$
**Output:** $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta)$ likelihood for $\theta$;
```

As the first likelihood estimation function we define AUCDF (Area Under Cumulative Distribution Function) (see Algorithm 3), which can use any of the similarity measures defined above. Different sources θ produce different empirical cumulative distributions of similarities to $\vec{r}_*$. If we compare two empirical distribution functions CDF_1 and CDF_2 from two different sources θ_1 and θ_2 and if the $AUCDF_1 < AUCDF_2$ then sample realizations from source θ_1 are more similar to fixed realization $\vec{r}_*$ than the sample realizations from source θ_2. This is the primary reason why we use the value $1 - AUCDF$ to estimate source likelihood $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta)$. Here, we use the assumption that the following inequality for the area under the cumulative distribution functions holds:

$$\int_0^1 \hat{P}(\varphi(\vec{r}_*, \vec{R}_{\theta_1}) \geq x) \mathrm{d}x \geq \int_0^1 \hat{P}(\varphi(\vec{r}_*, \vec{R}_{\theta_2}) \geq x) \mathrm{d}x \tag{3.8}$$

when node θ_1 is more likely to produce the realization $\vec{r}_*$ than the node θ_2. Note, that this measure is very similar to the area under the receiver operating characteristic in signal detection theory and machine learning theory.

The AvgTopK algorithm 4 represents a variant of the previous estimation function, which uses only k highest values from the tail of the probability density function of the random variable $\varphi(\vec{r}_*, \vec{R}_\theta)$.

In each simulation, we calculate the similarity (φ) between realization $\vec{R}_{\theta,i}$ and observed realization $\vec{r}_*$. The estimate $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta)$ is the average score over top k highest similarities $\varphi(\vec{r}_*, \vec{R}_{\theta,i})$ in n simulations (the tail of PDF).

3.3 The Naive Bayes – heuristic independence-based method

Now, we show the method which does not use any similarity function but assumes the independence among node states. The conditional probability that the node k in the realization $\vec{r}_*$ is infected from a source θ is:

$$\hat{P}(\vec{r}_*(k)=1|\Theta=\theta)=\frac{m_k+\varepsilon}{n+\varepsilon},\forall k\in G, \tag{3.9}$$

where m_k is the number of times that the node k got infected from the total of n simulations $SIR(p,q)$ from the source node θ and ε is a smoothing factor. Smoothing factor ε is necessary to mitigate the problem of zero values, stemming from the finite number of simulations used to calculate $\hat{P}(\vec{r}_*(k)=1|\Theta=\theta)$.

Then, we define the estimator for the likelihood of observing realization $\vec{r}_*$ from a source node θ is:

$$\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta)=\prod_{\{k:\vec{r}_*(k)=1\}}\hat{P}(\vec{r}_*(k)=1|\Theta=\theta)\prod_{\{j:\vec{r}_*(j)=0\}}(1-\hat{P}(\vec{r}_*(j)=1|\Theta=\theta)). \tag{3.10}$$

This equation uses the probability estimates that nodes $\{k:\vec{r}_*(k)=1\}$ from realization $\vec{r}_*$ got infected and the probability estimates that nodes $\{j:\vec{r}_*(j)=0\}$ from realization $\vec{r}_*$ did not get infected from the source node θ.

Note that, the probability of finding an infected node k at time t is dependent on other infected nodes prior to time t. Nevertheless, we use the independence assumption to estimate the rank of potential sources. There is an obvious resemblance between this approach and the well known studied probabilistic classifier - Naive Bayes.

In order to have more stable numerical likelihood estimations, we used the log likelihood variant for estimating $\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta))$ (see Algorithm 5).

3.4 The Direct Monte Carlo – approximate method

In this section, we proceed with the construction of an approximation algorithm, which is able to approximate the theoretical source posterior probability distribution. Note, that the Naive Bayes, AvgTopK and AUCDF are heuristics algorithms which do not provide a guarantee for the accuracy of solution. For each node θ_i from the set S of the realization $\vec{r}_*$, a large number n of epidemic spreading simulations [46] with duration T is performed with θ_i as an epidemic source. The number of simulations n_i which coincides with the realization $\vec{r}_*$ is recorded. After the simulations for all potential nodes in the realization $\vec{r}_*$ are finished, the probability of the node θ_i being the source of the epidemic is calculated as $P(\Theta=\theta_i|\vec{R}=\vec{r}_*)=n_i/\sum_j n_j$.

If the size of the realization $\vec{r}_*$ is big, the number of simulations required to obtain reliable

Algorithm 5 Naive Bayes likelihood estimation function $(G, p, q, \vec{r}_*, T, \theta, n)$

```
**Input:** $G$ - network structure, $(p,q)$ - SIR process parameters, $\vec{r}_*$ - observed realization prior to some temporal threshold $T$, $\theta$ - initial source for which likelihood is calculated, $n$ - a number of simulations
- $m_k = 0 : \forall k \in V$ from $G$;
for $i$ = 1 to $n$ (number of simulations) do
    - Run SIR simulation $(p,q)$ with $\Theta = \theta$ and obtain realization $\vec{R}_{\theta,i}$ prior to the temporal threshold $T$;
    - Update: $m_k = m_k + 1$; $\forall k$ which was infected in $\vec{R}_{\theta,i}$;
end for
- Calculate:
    $\hat{P}(\vec{r}_*(k) = 1 | \Theta = \theta) = \frac{m_k + \varepsilon}{n + \varepsilon}, \forall k \in G$
- Calculate the log likelihood: $log(\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta)) =$
    $= \sum_{\{k : \vec{r}_*(k) = 1\}} log(\hat{P}(\vec{r}_*(k) = 1 | \Theta = \theta)) + \sum_{\{j : \vec{r}_*(j) = 0\}} log(1 - \hat{P}(\vec{r}_*(j) = 1 | \Theta = \theta));$
**Output:** $log(\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta))$ likelihood for $\theta$;
```

frequencies can be prohibitive large. As the estimation for different nodes is independent, the computations are done in parallel using a high performance Message Passing Library with the C++ language and the Igraph library [51]. Furthermore, we have employed a pruning mechanism for the SIR model, where the Monte Carlo epidemic simulation is stopped at the time step $t < T$ if the current simulated realization has infected a node which has not been infected in $\vec{r}_*$. This pruning mechanism provides a substantial acceleration and, what is more important, does not induce any errors in our estimation. Note, that the direct Monte Carlo has a slow convergence and therefore we use it only for finding the exact solutions on small-size benchmark networks.

The accuracy of direct Monte-Carlo approximations are controlled by the convergence conditions. We estimate two source PDFs, one (P_i^n) by doing n, and the other (P_i^{2n}) with $2n$ independent simulations, where n is usually in the range $10^6 - 10^9$. Then, we choose the ML node as the node with the highest probability in (P_i^{2n}). The PDFs which satisfies the two following conditions:

$$|P_{ML}^{2n} - P_{ML}^{n}| / P_{ML}^{2n} \leq c, |P_i^n - P_i^{2n}| \leq c : \forall i \in V, \tag{3.11}$$

are said to converge with the direct Monte Carlo method. The relative error convergence with the value c on all the nodes is a too strict computational condition for practical purposes as we are interested in finding the high probability source nodes.

Here we devise the rule for pruning realizations at some temporal point $t < T$ whose contribution is zero. Let us also define the error term for every simulated realization $\vec{r}_i^{\,t}$ at some point

Algorithm 6 Direct Monte Carlo likelihood estimation function $(G, p, q, \vec{r}_*, T, \theta, n)$

```
Input: $G$ - network structure, $(p,q)$ - SIR process parameters, $\vec{r}_*$ - observed realization prior to some temporal threshold $T$, $\theta$ - initial source for which likelihood is calculated, $n$ - a number of simulations
$n_\theta = 0$;
for $i$ = 1 to $n$ (number of simulations) do
  for $t$ = 1 to $T$ (simulation steps) do
    Run SIR simulation $(p,q,\theta)$ iteration for time step $t$ and obtain $\vec{r_i^t}$
    Calculate error term:
      $\varepsilon_t(\vec{r_i^t}, \vec{r_*^T}) = \frac{1}{N} \sum_{k \in V} \psi_\wedge(1 - \vec{r}_*{}^T(k), \vec{r}_i{}^t(k)).$
    if $\varepsilon_t(\vec{r_i^t}, \vec{r_*^T}) > 0$ then
      break; (stop SIR simulation at time $t$)
    end if
  end for
  if $\vec{r_i^t}$ equals $\vec{r}_*$ then
    update $n_\theta = n_\theta + 1$
  end if
end for
Calculate: $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta) = \frac{n_\theta}{n}$;
Output: $\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta)$ likelihood for $\theta$;
```

in time $t < T$ w.r.t observed realization $\vec{r_*^T}$ at time T:

$$\varepsilon_t(\vec{r_i^t}, \vec{r_*^T}) = \frac{1}{N} \sum_{k \in V} \psi_\wedge(1 - \vec{r}_*{}^T(k), \vec{r}_i{}^t(k)), \tag{3.12}$$

where $\psi_\wedge(x_1, x_2)$ function is defined as the binary "AND" function:

$$\psi_\wedge(x_1, x_2) = \begin{cases} 1 & : (x_1 = 1 \text{ and } x_2 = 1), \\ 0 & : \text{else}. \end{cases} \tag{3.13}$$

Note, that the value $1 - \vec{r}_*{}^T(k)$ equals 1 when the node k is susceptible ($\vec{r}_*{}^T(k) = 0$) and equals 0 otherwise. Therefore, this term $\psi_\wedge(1 - \vec{r}_*{}^T(k), \vec{r}_i{}^t(k))$ captures the following events: the node k is susceptible in observed realization $\vec{r}_*$ and at the same time is infected in simulated realization $\vec{r}_i$.

The error term calculates the number of corresponding nodes, which are non-infected in the observed realization $\vec{r_*^T}$ at time T and infected in the simulated realization $\vec{r_i^t}$ at time t.

Proposition 3.4.1. *Monte Carlo SIR realization simulation $\vec{r_i^t}$ at time $t < T$ can be terminated if the $\varepsilon_t(\vec{r_i^t}, \vec{r_*^T}) > 0$ and it will have no effect on the final estimation.*

Proof. If at time t the error term $\varepsilon_t(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > 0$, then at time T the error can only increase: $\varepsilon_T(\vec{r}_i^{\,T}, \vec{r}_*^{\,T}) \geq \varepsilon_t(\vec{r}_i^{\,t}, \vec{r}_*^{\,T})$ because the error term $\varepsilon_t(\vec{r}_i^{\,t}, \vec{r}_*^{\,T})$ is monotonic increasing function w.r.t. time $t+1, t+2, \ldots, T$ and direct Monte Carlo estimator rejects any realization with the positive error term: $\varepsilon_T(\vec{r}_i^{\,T}, \vec{r}_*^{\,T}) > 0$. The infected state $(\vec{r}_i^{\,t}(k) = 1)$ is absorbing state w.r.t time t. Once the node leaves the susceptible state it cannot come back to it in the SIR model. ☐

The pruning mechanism provides a substantial acceleration (see Figure 3.2) without inducing any errors to our estimation.

Note, that the solutions from direct Monte-Carlo algorithm have been compared with the exact analytical combinatoric method [58] on small benchmark example and they show excellent agreement (for more information see supplementary materials in original article [58]). The analytical combinatoric method assigns to each node of degree n a generating function which is maximally $(n+1)$-dimensional, which captures the events of node first infection and infection spreading through its edges at specific times. Then, by multiplication of the generating functions of all the infected nodes from a realization, we are able to merge all contributions together and get the source probability distribution. A serious disadvantage of the analytical method is that the calculations become prohibitively intricate in the case of non tree-like configurations.

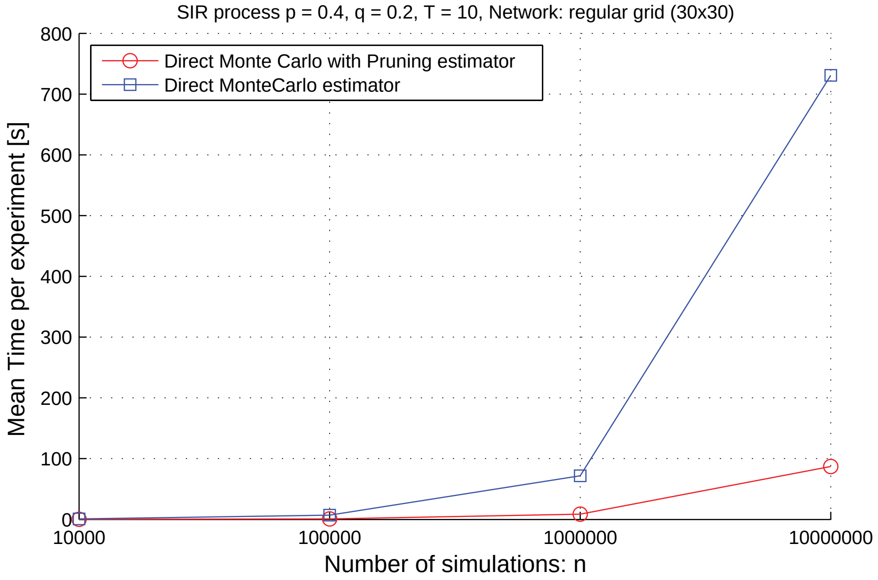


Figure 3.2: The speedup of direct Monte Carlo estimation with the pruning rule for experiment with n simulations per source node. Comparison of run-time per source detection experiment on 30 cpu cores (4 x AMD Opteron Processor 6134, 2.3 GHz with 8 cores each) averaged over 10 experiments.

3.5 The Soft Margin – approximate relaxed method

We proceed with the construction of an estimator which is able to find an approximate source probability distribution, but which is computationally much more efficient than the direct Monte Carlo or the analytical combinatoric method [58]. We first need to introduce some useful definitions. The i-th sample of the random variable $\vec{R}_\theta$ is denoted with $\vec{r}_{\theta,i}$ and is obtained using the Monte Carlo simulation of the contagion process with the θ as the source. As a similarity measure $\varphi : (R^N \times R^N) \to [0,1]$, we use the Jaccard similarity function calculated as the ratio of the size of the intersection of sets $\vec{r}_1$ and $\vec{r}_2$ and the size of their union. The random variable $\varphi(\vec{r}_*, \vec{R}_\theta)$ measures the similarity between a fixed realization $\vec{r}_*$ and a random realization that comes from the SIR process with the source θ. The cumulative distribution function of the random variable $\varphi(\vec{r}_*, \vec{R}_\theta)$ is denoted $F_\theta(x)$, where x is the value the similarity variable. By taking the derivative of the unbiased estimator of $\hat{F}_\theta(x)$ from n samples we get the PDF of $\varphi(\vec{r}_*, \vec{R}_\theta)$:

$$\hat{f}_\theta(x) = \frac{d}{dx}\hat{F}_\theta(x) = \frac{1}{n}\sum_{i=1}^{n} \delta\left(x - \varphi(\vec{r}_*, \vec{r}_{\theta,i})\right), \tag{3.14}$$

where $\delta(x)$ denotes the Dirac delta distribution.

Now, we define **the Soft Margin estimator** with the following formula:

$$\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta) = \int_0^1 w_a(x) f_\theta(x) \mathrm{d}x, \tag{3.15}$$

where $w_a(x)$ is a weighting function and $f_\theta(x)$ is the PDF function of the random variable $\varphi(\vec{r}_*, \vec{R}_\theta)$. We use the following Gaussian weighting form: $w_a(x) = e^{-(x-1)^2/a^2}$. In the limit where the parameter $a \to 0$, we obtain the unbiased estimate (direct Monte Carlo method) of the likelihood $P(\vec{R} = \vec{r}_* | \Theta = \theta)$. For cases when the parameter $a > 0$, we obtain an estimator which estimates the likelihood by using the Soft Margin function $w_a(x)$ to accept a contribution of a specific realization $\vec{r}_{\theta,i}$, contrary to the unbiased estimate ($a = 0$) which rejects the contribution of all realizations that are not exactly the same as the observed realization $\vec{r}_*$. The motivation for the Soft Margin was the following: we turn the problem of choosing the realization with the similarity $\varphi = 1$ to the problem of choosing realizations with the similarity in the interval where the contributions drops with the Gaussian function $w_a(x)$ from the point $\varphi = 1$.

We can simplify the Soft Margin formula:

$$\hat{P}(\vec{R} = \vec{r}_* | \Theta = \theta) = \int_0^1 w_a(x) \hat{f}_\theta(x) \mathrm{d}x = \int_0^1 w_a(x) \underbrace{\frac{1}{n}\sum_{i=1}^{n} \delta\left(x - \varphi(\vec{r}_*, \vec{r}_{\theta,i})\right)}_{\hat{f}_\theta(x)} \mathrm{d}x, \tag{3.16}$$

by using the property of delta distribution: $\int_{-\infty}^{\infty} f(x)\delta(x-b)\mathrm{d}x = f(b)$,

$$\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta) = \frac{1}{n}\sum_{i=1}^{n}\int_0^1 w_a(x)\delta\left(x-\varphi(\vec{r}_*,\vec{r}_{\theta,i})\right)\mathrm{d}x = \frac{1}{n}\sum_{i=1}^{n} w_a(\varphi(\vec{r}_*,\vec{r}_{\theta,i})). \tag{3.17}$$

Remember that all the sample values $\varphi(\vec{r}_*,\vec{r}_{\theta,i})$ are in interval $[0,1]$, so we have added small ε error to capture samples $\vec{r}_{\theta,i}$ whose similarity is not exactly $\varphi(\vec{r}_*,\vec{r}_{\theta,i}) = 1$.
We use the following weighting form: $w_a(x) = e^{-(x-1)^2/a^2}$ and thus the likelihood estimation is equal to:

$$\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta) = \frac{1}{n}\sum_{i=1}^{n} e^{\frac{-(\varphi(\vec{r}_*,\vec{r}_{\theta,i})-1)^2}{a^2}}. \tag{3.18}$$

Algorithm 7 Soft Margin likelihood estimation function $(G, p, q, \vec{r}_*, T, \theta, n)$

```
Input: $G$ - network structure, $(p,q)$ - SIR process parameters, $\vec{r}_*$ - observed realization prior to some temporal threshold $T$, $\theta$ - source for which likelihood is calculated, $n$ - a number of simulations
for $i$ = 1 to $n$ (number of simulations) do
    Run SIR simulation $(p,q,\theta)$ and obtain epidemic realization at time $T$: $\vec{R}_{\theta,i}$;
    Calculate and save $\varphi_i = \varphi(\vec{r}_*,\vec{R}_{\theta,i})$ ;
end for
Calculate likelihood:
    $\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta) = \frac{1}{n}\sum_{i=1}^{n} e^{\frac{-(\varphi_i-1)^2}{a^2}}$.
Output: $\hat{P}(\vec{R}=\vec{r}_*|\Theta=\theta)$ likelihood for $\theta$;
```

Finally, we do not need to set the Soft Margin width parameter a in advance. After we calculate the estimated PDF for every potential source $\hat{F}_\theta(x)$, we can choose the parameter a as the infimum of the set of parameters for which the PDFs have converged. For example, after we calculate $\hat{f}_\theta(x)$ for every potential source, we recalculate the source probability distribution for different values of parameter a in range: $\{1/2, (1/2)^2, (1/2)^3, ..., (1/2)^{15}\}$. Then, we measure the convergence property of estimated PDFs: $\hat{P}_a^n(\Theta=\theta_i|\vec{R}=\vec{r}_*)$ for different values of Soft Margin weight a and different number of simulations n. We use the following convergence condition for the source PDFs: $|\hat{P}_a^n(\Theta=\theta_{MAP}|\vec{R}=\vec{r}_*) - \hat{P}_a^{2n}(\Theta=\theta_{MAP}|\vec{R}=\vec{r}_*)| \leq 0.05$, where θ_{MAP} is the node i with the maximum estimated source probability in $\hat{P}_a^{2n}(\Theta=\theta_i|\vec{R}=\vec{r}_*)$. The smaller the parameter a, the estimations become more similar to the direct Monte Carlo estimator if the PDFs have converged. Note, that the maximum likelihood (ML) node is the same as the maximum posteriori (MAP) node if our prior source probabilities are equal. As the computational complexity of calculating $\hat{f}_\theta(x)$ is a few orders of magnitude higher than the complexity of recalculating source PDF for different parameters a, one does not need to set parameter a in advance, but rather choose the near-optimal value of a for a specific number of simulations n.

Relaxation of parameters

Up to this point we have assumed that the epidemic duration T or the starting point t_0 were given in advance. Strictly speaking, we should have written the parameter T in all the conditional probabilities $P(\vec{R} = r_*|\Theta = \theta, T = t - t_0)$ instead of just $P(\vec{R} = r_*|\Theta = \theta)$, but this would just complicate the notation since we could also put other given parameters like G, p and q in the condition. Instead, unless otherwise stated, we assume that parameters: T, G, p and q are given. Now we explain how to relax the assumption on specific epidemic parameters. For example, if we know the time t when the realization was observed, but the epidemic starting moment t_0 is not known in advance, by marginalization over all possible t_0 outcomes we get:

$$P(\vec{R} = \vec{r}_*|\Theta = \theta) = \sum_{t_0=0}^{t} P(\vec{R} = \vec{r}_*, T = t - t_0|\Theta = \theta), \tag{3.19}$$

where the variable T denotes the epidemic duration. This expression can be further transformed into:

$$P(\vec{R} = \vec{r}_*|\Theta = \theta) = \sum_{t_0=0}^{t} P(\vec{R} = \vec{r}_*|\Theta = \theta, T = t - t_0) P(T = t - t_0|\Theta = \theta). \tag{3.20}$$

Now, the term $P(\vec{R} = \vec{r}_*|\Theta = \theta, T = t - t_0)$ can be calculated with the Soft Margin estimator like before, and the term $P(T = t - t_0|\Theta = \theta)$ denotes the prior distribution of epidemic duration or epidemic start. But, we do not estimate $\hat{P}(\vec{R} = \vec{r}_*|\Theta = \theta)$ by definition due to its computational cost, but rather by another sample estimation. First, we sample a T_i from the prior probability distribution $P(T = t - t_0|\Theta = \theta)$ and then obtain the sample realization $\vec{r}_{\theta,i}$ for a given T_i. We repeat the procedure n times, obtain n temporal samples $\{T_1, ..., T_n\}$ and obtain n corresponding realizations $\{\vec{r}_1, ..., \vec{r}_n\}$. Then, we estimate $\hat{P}(\vec{R} = r_*|\Theta = \theta)$ with the Soft Margin estimator from n realizations: $\{\vec{r}_1, ..., \vec{r}_n\}$ from their similarities to the observed realization: $\{\varphi(\vec{r}_*, \vec{r}_{\theta,1}), ... \varphi(\vec{r}_*, \vec{r}_{\theta,n})\}$.

$$\hat{P}(\vec{R} = r_*|\Theta = \theta) = \frac{1}{n}\sum_{i=1}^{n} e^{\frac{-(\varphi(\vec{r}_*, \vec{r}_{\theta,i})-1)^2}{a^2}}.$$

This is the sample estimation, because we can regroup realizations with the same T and get:

$$\hat{P}(\vec{R} = r_*|\Theta = \theta) = \sum_{T} \hat{P}(\vec{R} = r_*|\Theta = \theta, T) \underbrace{\hat{P}(T|\Theta = \theta)}_{\frac{k_T}{n}}$$

$$\hat{P}(\vec{R} = r_*|\Theta = \theta) = \sum_{T} \underbrace{\frac{1}{k_T} \sum_{j:T(j)=T} e^{\frac{-(\varphi(\vec{r}_*, \vec{r}_{\theta,j})-1)^2}{a^2}}}_{\textit{Soft Margin}} \frac{k_T}{n} = \frac{1}{n}\sum_{i=1}^{n} e^{\frac{-(\varphi(\vec{r}_*, \vec{r}_{\theta,i})-1)^2}{a^2}}.$$

Pruning and Soft margin

The Soft margin estimator uses the following realization similarity weighting function: $w_a(x) = e^{-(x-1)^2/a^2}$. Now, if we have the following weighting function $w_a^{'}(x)$, that has a cutoff at point $x = \varphi_B$ defined with the following formula:

$$w_a^{'}(x) = \begin{cases} w_a(x) = e^{-(x-1)^2/a^2} & : x \geq \varphi_B, \\ 0 & : x < \varphi_B. \end{cases}, \tag{3.21}$$

then we can introduce the pruning mechanism for terminating the simulation at time t before the stopping point T with no effect on the likelihood estimation with the weighting function $w_a^{'}(x)$.

Let us recall the definition of the error term for every simulated realization $\vec{r}_i^t$ at some point in time $t < T$ w.r.t observed realization $\vec{r}_*^T$ at time T:

$$\varepsilon_t^B(\vec{r}_i^t, \vec{r}_*^T) = \frac{1}{N} \sum_{k \in V} \psi_\wedge(1 - \vec{r}_*{}^T(k), \vec{r}_i{}^t(k)), \tag{3.22}$$

where $\psi_\wedge(x_1, x_2)$ function is defined as the binary "AND" function:

$$\psi_\wedge(x_1, x_2) = \begin{cases} 1 & : (x_1 = 1 \text{ and } x_2 = 1), \\ 0 & : \text{else}. \end{cases} \tag{3.23}$$

Lemma 3.5.1. *The error term $\varepsilon_t^B(\vec{r}_i^t, \vec{r}_*^T)$ is monotonic increasing function w.r.t. time $t+1, t+2, ..., T$.*

Proof. Once the node goes from susceptible state $\vec{r}_i^t(k) = 0$ at time t, it cannot come back for any time after t in the SIR model or any other compartmental model with no recurrent states. □

Proposition 3.5.1. *The contribution of all realizations $\vec{r}_i^t$ at time $t < T$ to the likelihood estimation at time T is zero if the $\varepsilon_t^B(\vec{r}_i^t, \vec{r}_*^T) > (1 - \varphi_B)$, where the term $(1 - \varphi_B)$ denotes the maximal error term we can have.*

Proof. We need to prove that $\varepsilon_t^B(\vec{r}_i^t, \vec{r}_*^T) > (1 - \varphi_B)$ implies $\varphi_i^T < \varphi_B$, where φ_i^T denotes the similarity of realization $\vec{r}_i^T$ at time T. The connection between error terms and (XNOR) similarity is the following:

$$N(1 - \varphi_i^t) = N - \sum_{k \in V} \psi_\oplus(\vec{r}_*{}^T(k), \vec{r}_i{}^t(k))$$

Recall the definition of the XNOR $\psi_{\oplus}(x_1, x_2)$ function:

$$\psi_{\oplus}(x_1, x_2) = \begin{cases} 1 & : (x_1 = 1 \text{ and } x_2 = 1) \text{ or } (x_1 = 0 \text{ and } x_2 = 0)\ , \\ 0 & : \text{else.} \end{cases} \tag{3.24}$$

$$N(1-\varphi_i^t) = \sum_{k \in V} \psi_{\wedge}(\vec{r}_*{}^T(k), 1-\vec{r}_i{}^t(k)) + \sum_{k \in V} \psi_{\wedge}(1-\vec{r}_*{}^T(k), \vec{r}_i{}^t(k))$$

$$\implies (1-\varphi_i^t) = \underbrace{\frac{1}{N}\sum_{k \in V} \psi_{\wedge}(\vec{r}_*{}^T(k), 1-\vec{r}_i{}^t(k))}_{\varepsilon_t^A} + \underbrace{\frac{1}{N}\sum_{k \in V} \psi_{\wedge}(1-\vec{r}_*{}^T(k), \vec{r}_i{}^t(k))}_{\varepsilon_t^B}$$

$$(1-\varphi_i^t) = \varepsilon_t^A(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) + \varepsilon_t^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T})$$

Therefore, the total error $1-\varphi_i^t$ has two components: ε_t^A and ε_t^B. Now, if we assume that $\varepsilon_t^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > (1-\varphi_B)$ from previous lemma, we conclude that $\varepsilon_T^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > (1-\varphi_B)$ and if we add non-negative error term to the left side of previous inequality we get:

$$\varepsilon_T^A(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) + \varepsilon_T^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > (1-\varphi_B) \implies (1-\varphi_i^T) > (1-\varphi_B) \implies \varphi_i^T < \varphi_B.$$

Therefore, if at time t the error term $\varepsilon_t^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > (1-\varphi_B)$ then at time T the similarity φ_i^T can only be lower than the border cutoff value φ_B. ☐

Corollary 3.5.1. *Monte Carlo SIR realization simulation $\vec{r}_i^{\,t}$ at time $t < T$ can be terminated if the $\varepsilon_t^B(\vec{r}_i^{\,t}, \vec{r}_*^{\,T}) > (1-\varphi_B)$ as it will have no contribution to the likelihood calculated by weighted function $w_a^{'}(x)$ with a cutoff at φ_B.*

3.6 Benchmark analysis

In order to do a proper comparison of different source detection estimators, there has to exist a proper measure of the quality of solution. Because of the non-uniqueness of a single source node we will not compare the estimators by their ability to detect the true source, but instead by comparing their estimated source probabilities to the source probability distribution of the ideal solution. We can generate approximation of the ideal solution by using the Direct Monte-Carlo estimator with strong convergence conditions.

We have generated a series of benchmark cases for which we have calculated the probability distributions over the potential source candidates using the direct Monte Carlo estimator. Note that the direct Monte Carlo estimator has been validated by comparison with the analytical combinatoric solution [58]. In order to be sure that the direct Monte Carlo estimator outputs valid results on realizations with cycles, we set its convergence condition to $c = 0.05$. The convergence condition for the direct Monte Carlo solution at a $2x$ number of simulations was set to be: ML node relative error is: $|P_{ML}^{2x} - P_{ML}^{x}|/P_{ML}^{2x} \leq c$ and the maximal absolute error for all other nodes is: $|P_i^{2x} - P_i^{x}| \leq c$.

We have used a small 4-connected lattice ($N = 30 \times 30$) and SIR processes with different parameters (p, q, T) with the direct Monte Carlo estimator with $[10^6 - 10^8]$ simulations per source, depending on the convergence condition, to obtain the source PDFs for our benchmark.

Next, we compare the representatives of three different classes of source detection estimators: network centrality estimators, belief propagation estimators and our Monte Carlo estimators. For the network centrality estimation, we use the Jordan estimator [36], which assigns a weight to each potential node candidate, which is equal to the maximal topological distance from the node candidate to all other infected nodes in a realization. Although the Jordan estimator uses a very simple rule, it outperforms most of other network centrality measures. The representative of the belief propagation estimators is the Dynamic Message Passing Algorithm (DMP) [25], which uses a mean-field-like approximation (independence approximation) about the node states along with a recursive analytical formula for the treelike networks to estimate the source likelihoods.

Finally, we use our Soft Margin estimator which falls into the general class of the Monte Carlo estimators. Note that, when comparing the Soft Margin estimator, we evaluate it with a few orders of simulations less than the number of simulations used to generate the benchmark standard solution. The maximum likelihood (ML) node for each realization is determined by the benchmark solution. Then, for each estimator, we measure the ML accuracy performance and ML probability estimate error. The ML accuracy measures the expected number of times in which the estimators rank the ML node on rank 1 and relative ML error measures the ability to reconstruct the associated probability for the ML node. In Figure 3.3, we can see the mean relative errors and the accuracy of the ML node for different estimators. From this analysis, we observe that most estimators are trying to produce a valid ranking (ML accuracy) but without estimating the true probability (ML relative error). The exception is the Soft Margin estimator, which estimates both the valid ranking and a valid source probability at the same time. The source probability distribution for the observed realization contains more information about the initial conditions than just the ranking of potential candidates, especially for cases where the detectability limits are more pronounced.

Here, we provide the comparison of different estimators for the SIR model w.r.t. ML probability relative errors (see Figure 3.5) and ML accuracy (see Figure 3.6) on the benchmark dataset. The correct solutions were calculated with the direct Monte Carlo estimator with $[10^6 - 10^8]$ simulations per source depending on convergence condition. The convergence condition for the direct Monte Carlo solution at a $2x$ number of simulations was set to be: ML node relative error is: $|P_{ML}^{2x} - P_{ML}^{x}|/P_{ML}^{2x} \leq 0.05$ and the maximal absolute error for all other nodes is:

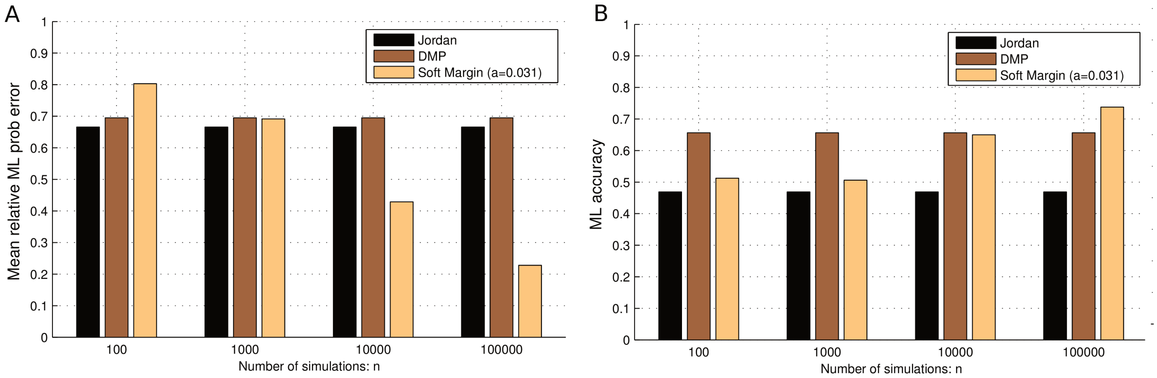


Figure 3.3: Comparison of different estimators (Network centrality-Jordan, Belief Propagation-DMP and Monte Carlo-Soft Margin) performance with the ML relative probability estimation error (plot A) and ML accuracy (plot B) with the 160 different benchmark cases. Benchmark cases were calculated on a small 4-connected lattice with ($N = 30 \times 30$) and SIR process with different parameters (p, q, T) with the direct Monte Carlo estimator with $[10^6 - 10^8]$ simulations per source depending on convergence condition with $c = 0.05$.

$|P_i^x - P_i^{2x}| \leq 0.05$. We have compared the centrality-like estimators: Distance [40] and Jordan [36] centrality, Belief propagation estimator: DMP [25], different Monte Carlo estimators: the AUCDF, the AvgTopK and the NaiveBayes [39] and two baseline solutions: Rnd (source likelihood is a random number from [0,1]) and Const (all sources are equal).
Most of the aforementioned estimators do not output explicit source probability distribution function, but rather a ranking list with appropriate weights w_i, from which we calculate the source PDF by re-normalization with factor $\sum_j w_j$ to get a PDF. We have used our implementation of distance [40], Jordan [36] centrality and Belief propagation estimator DMP [25].

In this thesis, we use a conservative information about the node state at observed moment t, we only observe whether the node is susceptible or not (realization $\vec{R}$ is a binary vector). This implies that we do not need additional information to distinguish whether the node state is recovered or it is still infective. The original DMP [25] estimator additionally assumes that one can distinguish the recovered from the infective state. Therefore, in order to apply the DMP [25] algorithm to our scenario, we had to adopt the estimation formula so that the probability of the node being infected is merged with a probability of the node being recovered in order to estimate the probability of being in either Infective or Recovered compartment. All other calculations were implemented according to the original algorithm [25]. In order to verify our implementation of the DMP algorithm, we have compared our DMP implementation on tree network, where the node state probability estimation should be correct. We have measured the difference between the probability estimate that the node is susceptible after T steps with the DMP and the SIR Monte Carlo simulation algorithm and we observe that less than 1 % of nodes have the relative error greater than 0.001, which means that the SIR Monte Carlo simulation algorithm estimates are very close to DMP on tree networks (see Figure 3.4).

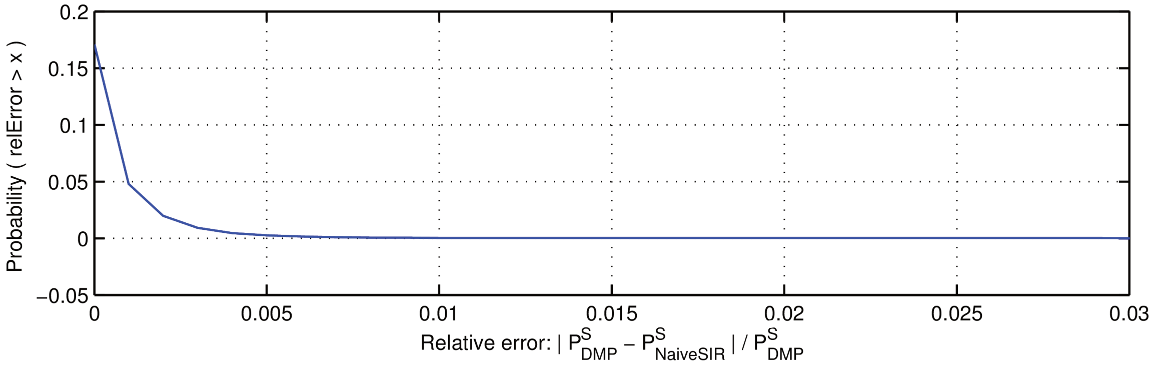


Figure 3.4: Comparing Dynamic Message Passing (DMP) estimates of the node state probability for $p = 0.3, q = 0.5, T = 10$ with the SIR simulation estimates (NaiveSIR) on the Albert-Barabashi tree network ($N = 5000, m_0 = 2, m = 1$). Distribution of relative errors of node being susceptible with SIR simulation ($n = 10^4$) w.r.t. DMP on tree network.

In a limit where the parameter $a \to 0$, for the Soft Margin estimator we obtain the unbiased estimate of the likelihood $P(\vec{R} = \vec{r}_* | \Theta = \theta)$. In cases when the parameter $a > 0$ we obtain an estimate which finds the likelihood by using the tail of PDF function $f(x)$ in a way that it uses the values of slightly different realizations to get estimate for observed realization $\vec{r}_*$. In Figure 3.7 plot A and B, we can see the effect of different Soft Margin widths a on the convergence. As the soft margin width parameter a decreases, it becomes more similar to the unbiased estimator, but the convergence becomes slower.

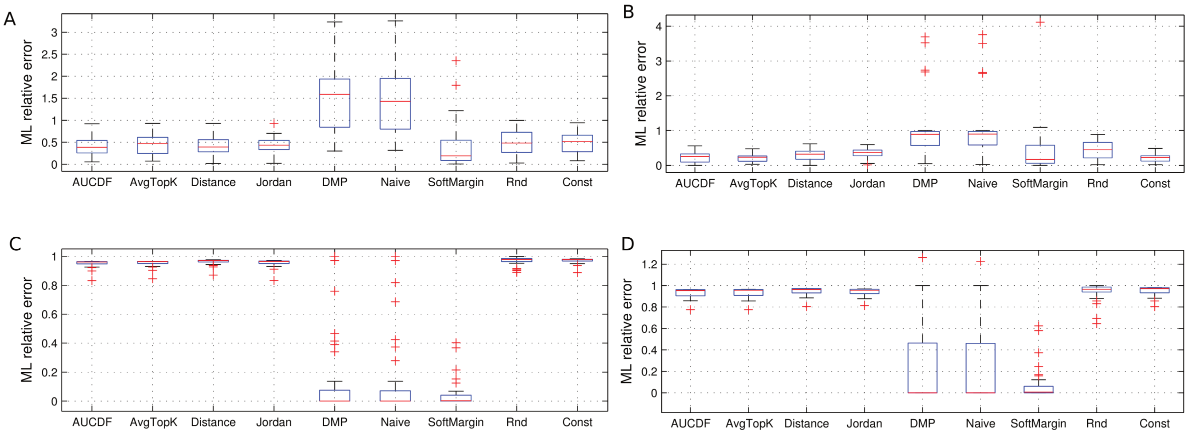


Figure 3.5: The comparison of maximum likelihood probability errors with box-plots for different estimators and soft margin with $a = 0.031$. The error is relative error of maximum likelihood estimation w.r.t. gold standard ML probability obtained with the direct Monte Carlo method for different parameters: $A = (p = 0.3, q = 0.3, T = 5)$, $B = (p = 0.3, q = 0.7, T = 5)$, $C = (p = 0.7, q = 0.3, T = 5)$ and $D = (p = 0.7, q = 0.7, T = 5)$.

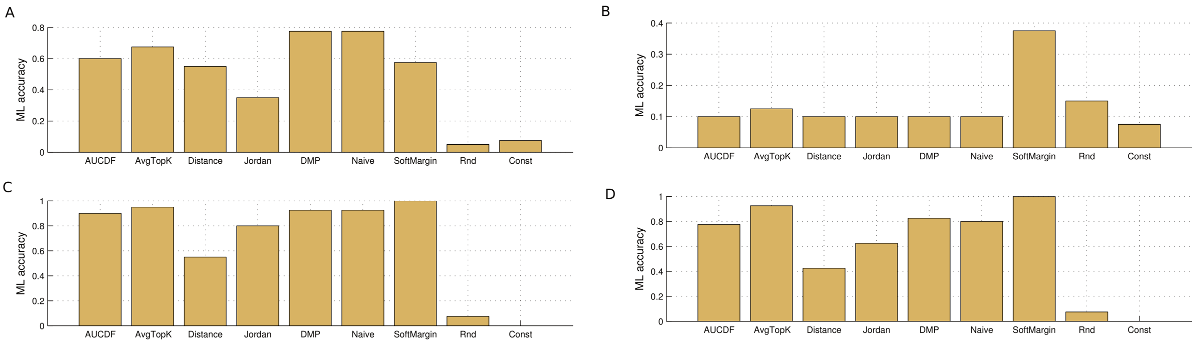


Figure 3.6: The comparison of accuracy of detecting the maximum likelihood node with mean accuracy for different estimators and soft margin with $a = 0.031$. ML accuracy is the ratio of how many times the estimator ranks the ML node on rank 1 and total number of trials (ranking measure). plot A-D correspond to different parameters: $A = (p = 0.3, q = 0.3, T = 5)$, $B = (p = 0.3, q = 0.7, T = 5)$, $C = (p = 0.7, q = 0.3, T = 5)$ and $D = (p = 0.7, q = 0.7, T = 5)$.

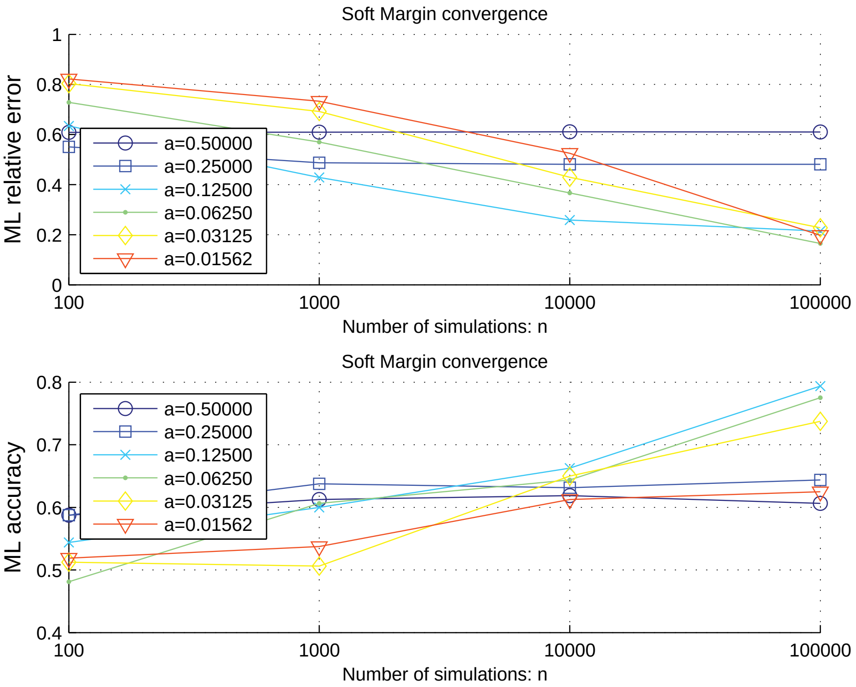


Figure 3.7: Comparison of Soft Margin estimators with different weights a with respect to the Maximum Likelihood relative probability estimation error (plot A) and Maximum Likelihood accuracy (plot B) using the average over 160 different benchmark cases. Benchmark cases were calculated on a small regular network (4-connected grid $N = 30 \times 30$) and SIR process with different parameters (p, q, T) with the direct Monte Carlo estimator with $[10^6 - 10^8]$ simulations per source depending on convergence condition.

3.7 Time complexity of estimators

The average run-time complexity $\overline{RT}$ of source detection Monte Carlo estimators (AUCDF, AvgTopk, Naive Bayes, Soft Margin, Direct Monte Carlo) is:

$$\overline{RT} \propto m \times n \times \overline{RT}_M,$$

where the term m denotes the number of potential sources in the observed realization, the term n denotes number of samples of the random variable $\vec{R}_\theta$ or alternatively the number of simulations of a contagion process and $\overline{RT}_M$ denotes the average run-time complexity of sampling one realization from contagion process M. Sampling the realizations from a contagion process in our case is equal to one Monte Carlo simulation of stochastic contagion model and returning one realization vector $\vec{r}_{\theta,i}$. Different Monte Carlo estimators (AUCDF, AvgTopk, Naive Bayes, Soft Margin, Direct Monte Carlo) have different convergence properties with respect to the number of samples n (see Section 3.6).

Note that in the worst-case scenario the number of potential sources is proportional to the network size $|\vec{r}_*| \propto N$, but in reality we are mostly interested in source detection problem when the number of potential sources is much smaller than the network size.

Note, that the calculations of likelihood for different sources θ in $\vec{r}_*$ are computed in a scalable parallel way with the MapReduce paradigm. The "Map" step distributes the source independent problems to worker nodes and "Reduce" step collects likelihood estimators and provides source probability distribution.

In the case when contagion process is the SIR model on an arbitrary static network, the average run-time complexity for the single SIR discrete simulation (NaiveSIR algorithm [46]) is:

$$\overline{RT}_{M1} \propto E(X_T) \times \bar{k} \times T,$$

where the term $E(X_T)$ denotes the expected number of infected nodes up to temporal threshold T and $\bar{k}$ is the average node degree.

In the case when the contagion process is the SIR model on a temporal network, the run-time complexity for the single SIR discrete simulation is:

$$\overline{RT}_{M2} \propto L_T,$$

Where L_T denotes the number of interactions during the epidemic process with duration T.

Note that after we have calculated the estimated PDF for each potential candidate node $\hat{f}_\theta(x)$, we can estimate source probabilities for different weight parameters a since this step is far less demanding than the previous steps.

Table 3.1: The source detection execution times on an empirical temporal network [59]. Execution times of source detection are measured with parallel computation on 50 CPU cores on the AMD Opteron(tm) Processor 6380, 2.5 GHz each. The computations are done in parallel by using a high performance Message Passing Library with the C++ language. Averaging was done over 50 independent experiments where the initial moment t_0 was chosen in period between $[100-200]$ days, the initial source was randomly selected from the set of active nodes in t_0 moment with the SIR STD model $(p = 0.3, q = 0.01)$ and realization $\vec{r}_*$ was observed at time $t = 300$ days. The run times are averaged over 50 independent experiments with the mean realization size $|\vec{r}_*|$ equal to 86. We have used the Soft Margin estimator, where the width parameter a was chosen as a minimum of parameters from the set: $\{1/2, (1/2)^2, (1/2)^3, ..., (1/2)^{15}\}$ for which the ML estimate converged up to 0.05 of relative change between consecutive simulations.

Number of simulations	n = 5000	n = 10000	n = 15000	n = 20000
Mean time [s]	2.9	5.8	8.7	11.6

3.8 Empirical temporal network of sexual contacts – a case study

Now, we demonstrate the applicability of our inference framework to detection of the source of the simulated STI epidemic spreading on an empirical temporal network of sexual contacts in Brazil (see Figure 3.8, plot E). We would like to note that this publicly available dataset [59] was obtained from the Brazilian Internet community, and it is used as an approximation of temporal sexual contacts. The data set consists out of the triplets (v_i, v_j, t), which represents the event that the nodes v_i and v_j had a sexual interaction at a time t. First 1000 days in the original data set are discarded due to the transient period with sparse encounters [59] and therefore all temporal moments are measured relative to the day 1000, identically to the authors [59] in the original study. In this case we use a temporal network with the SIR model of the STI ($p = 0.3$, $q = 0.01$). The upper limit of the transmission probability for the STI that was previously used on this contact network is $p = 0.3$ [59]. The recovery parameter $q = 0.01$ represents a disease with the mean recovery of 100 days.

Note that here the calculation of exact source probability distributions is computationally too demanding with the direct Monte Carlo or Analytic Combinatorics method. Therefore, we use the Soft Margin estimator with the smallest width a for which the ML node probability estimate converged and measure how well we can detect the true source. Our experiments consist of two parts: (i) simulation of STI spreading through a temporal network of sexual contacts and (ii) detection of the patient zero from the observed process. Realizations in (i) are generated using the STI model on the exact temporal contact network, where the patient zero is a randomly selected active node at a time point t_0 and epidemic observed at time t. In detection process (ii), we assume that we know the STI model parameters, but we relax the assumption of knowing the duration of the epidemic $T = t - t_0$ or the epidemic start moment t_0, exact ordering times of temporal contacts in the network and the whole realization vector $\vec{r}_*$ at time t.

The relaxation of knowing the starting point of the epidemic t_0 is done by using the marginalization over time, sampling over all possible starting points t_0 from a uniform probability distribution over $[t_0 - \varepsilon, t_0 + \varepsilon]$, $2\varepsilon = \{0, 50, 100\}$ days. In Figure 3.8 plot C, we show the summary results from 500 independent experiments, when the starting time t_0 was chosen from the interval of $[100 - 200]$ days, the end of the epidemic was set to the day $t = 300$ and we have used different uniform priors (ε) for the moment t_0. When the $\varepsilon = 50$ days of uniform uncertainty, we can still detect the source within its first neighbourhood (distance 0 and 1 from the source) in approximately 60% of experiments. These results are of great practical importance, since in reality we do not know the exact starting times, but rather only an upper and a lower bound on starting point.

Next, we demonstrate how the uncertainty in the temporal orderings of interactions within a time window of the length Δ affects the performance of source detection. We use a randomization algorithm which permutes time stamps inside of a bin of Δ days from the start to the end of the contact interaction network in a non-overlapping way. Therefore, this randomization permutes all time stamps that fall to the following temporal windows: $\{[0, \Delta - 1], [\Delta, 2\Delta - 1], ..., [k\Delta, t]\}$ of length Δ. The intuition behind this randomization is that, in reality, usually the data gathering procedure cannot guarantee ordering of temporal interactions smaller than some granularity of Δ days so that all orderings inside Δ days become equally likely. From Figure 3.8, plot D, we observe that higher uncertainty in orderings (higher Δ) reduces the detectability of the source of infection. However, the estimation framework is robust to small-scale interaction reordering.

Next, we show the results for source detection on temporal networks for contagion with the SIR model with high transmission probability $p = 0.8$ with recovery parameter $q = 0.05$ (expected recovery is 20 days). In Figure 3.9 plot A, we show the results, when starting t_0 was uniformly chosen from the interval $[100 - 200]$ day, the end of the epidemic was set to the day $t = 300$ and we used different uniform priors (ε) on t_0 moment. Plot B in the Figure 3.9 demonstrates the effect of detecting the source node from the network with randomized temporal ordering with parameter Δ.

In all the cases so far, we have assumed that we know the states of all the nodes in the network at the temporal snapshot t. Now, we will show that we can relax that assumption. We will assume that we can only observe the states of a random subset $O \subseteq V$ of all the nodes in the network. In Figure 3.10, we show the performance results for the source detection of STD disease when we know the states of 100%, 50% and 20 % of all the nodes in the network chosen randomly. Realization vectors $\vec{r}_*$ now can have the following values: $\{0, 1, ?\}$, where the "?" denotes the unknown state. In order to apply our methodology, we only need to adopt the similarity function in a way that it can handle the unknown states and determine the set of potential candidate sources S. Now, we use the same similarity function (Jaccard similarity), but we neglect the comparison with the missing state "?". The set of potential candidates is the union of all the nodes with state "1" and all the nodes with state "?" which are not surrounded with neighbours with "0" state only (they cannot be the initial source).

Disclaimer

The author of this thesis used the published existing dataset of sexual contacts in high-end prostitution because it contains valuable and rarely available information on temporal network of contacts serving as pathways of STD spreading. It is important to note that the use of this dataset does not reflect the author view, opinion and attitude on prostitution and it does not in any way imply that the author support the activities documented in the dataset or the way the data were gathered.

3.9 Empirical weighted network of air traffic – a case study

Now, we demonstrate the applicability on a socio-technical system of a diffusion of the infected agents on the world airport transportation system data[*] [60]. All we need is model M, which can simulate the spreading of disease (diffusion of the infected agents) on the global level of the air transportation system. In this model, each node represents an airport and each edge represents a connection where the diffusion of infected and susceptible travellers happen. The model M is parametrized by the P_{ij} probabilities, which represent the probability of diffusion along the edge. Under the assumption that the local infection dynamics is much faster than diffusion, this model represents the transmission of the disease on the global mobility network. One can further implement a more detailed model of meta-population spreading with the inclusion of adaptive factors to model M like quarantines, anti-viral drugs, mixing of infected and susceptible individuals inside cities, virus mutations, etc. In Figure 3.11 plot A, we observe one spatio-temporal realization of epidemic from Mexico City Juarez International airport with the SI diffusion model (p_{ij} from airport flux data, $T = 10$). The algorithm outputs top 5 maximum likelihood nodes (big red nodes) which are all near the Mexico City Juarez International airport and in Figure 3.11 plot B, we observe the performance on 100 independent experiments. This is an arbitrary example motivated by the recent H1N1 pandemic from Mexico in 2009. Although this example serves as proof of concept of a method the full scope of this applicability to the airport transportation network requires a much more detailed and extensive analysis.

---

[*] Official Airline Guide, http://www.oag.com

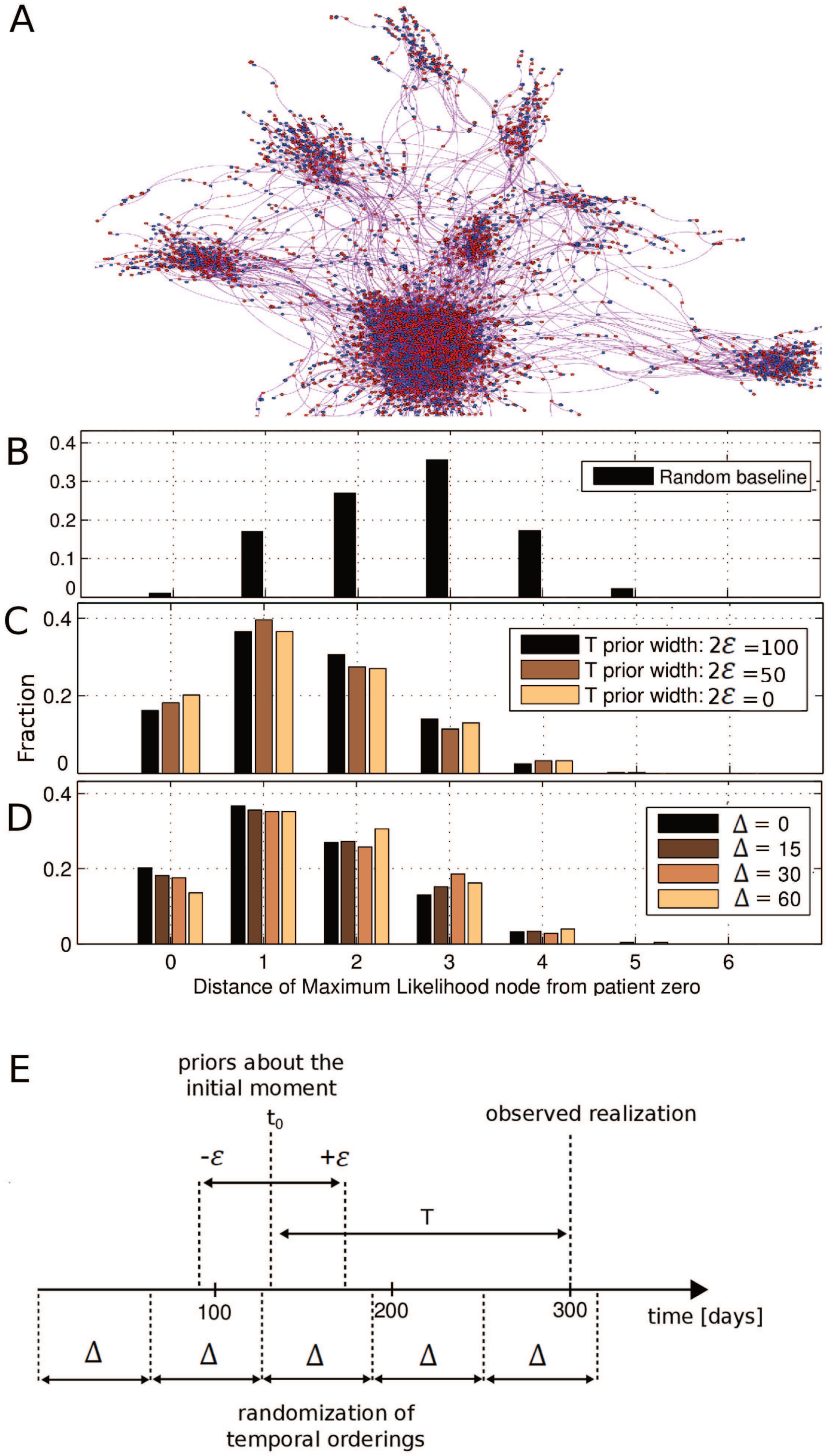


Figure 3.8: Sexually Transmitted Infections case study on the empirical temporal network.
Plot A: Visualization of aggregated empirical temporal network (≈ 3500 nodes) of sexual contacts in Brazil [59]. In plots B,C and D the performance is measured as the fraction of 500 experiments with specific graph distance of the maximum likelihood candidate to the true source. The average execution time of a single experiment to calculate source probability distribution over all potential candidates was around 12 seconds (on 50 cpu cores) with 20000 STI simulations per node. **Plot B**: The baseline performance of a random estimator, which assigns a random number between 0 and 1 as the likelihood for potential node candidates. **Plot C**: The influence of prior knowledge about initial outbreak moment $[t_0 - \varepsilon, t_0 + \varepsilon]$ of the outbreak on performance. **Plot D**: The influence of randomized temporal ordering of interactions within Δ days, with $\varepsilon = 0$ (we know the starting time t_0) on performance. **Plot E**: Diagram of temporal evolution of network with the experiment descriptor parameters (t_0,T,ε,Δ).

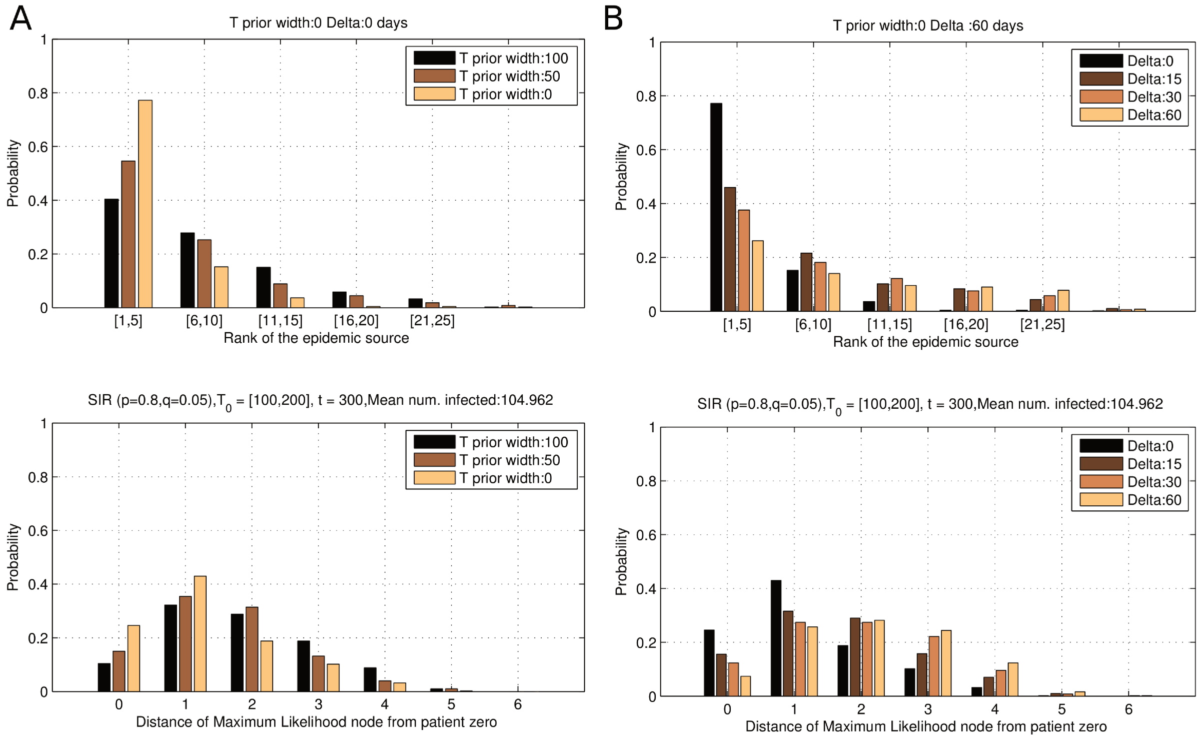


Figure 3.9: The Source detection of simulated sexually transmitted infections spreading in an empirical spatio-temporal network of sexual contacts in Brazil. The experiment consists of 500 experiments where the initial moment t_0 was uniformly chosen in period between $[100-200]$ days, the initial source was randomly selected from the set of active nodes at the moment t_0 with the SIR model $(p=0.8, q=0.05)$ and realization $\vec{r}_*$ was observed at time $t=300$ days. **Plot A**: The influence of prior knowledge about initial outbreak moment $[t_0-\varepsilon, t_0+\varepsilon]$. **Plot B**: The influence of detecting the source node from temporal networks with randomized temporal ordering of interactions within Δ days.

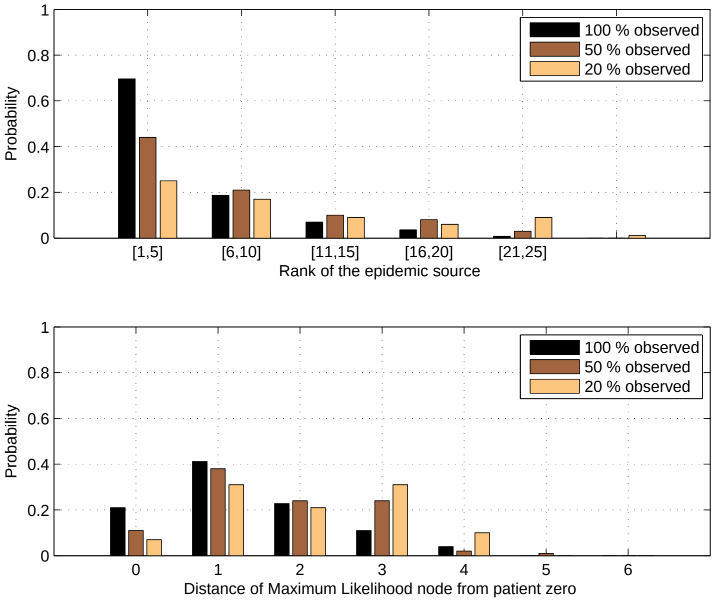


Figure 3.10: The Source detection of simulated sexually transmitted infections spreading in an empirical spatio-temporal network of sexual contacts in Brazil when we know the states of 100%, 50% and 20 % of all the nodes in the network chosen randomly. The experiment consists of 100 experiments where the initial moment t_0 was uniformly chosen in period between $[100-200]$ days, the initial source was randomly selected from the set of active nodes in the moment t_0 with the SIR model $(p=0.3, q=0.01)$ and realization $\vec{r}_*$ was observed at time $t=300$ days.

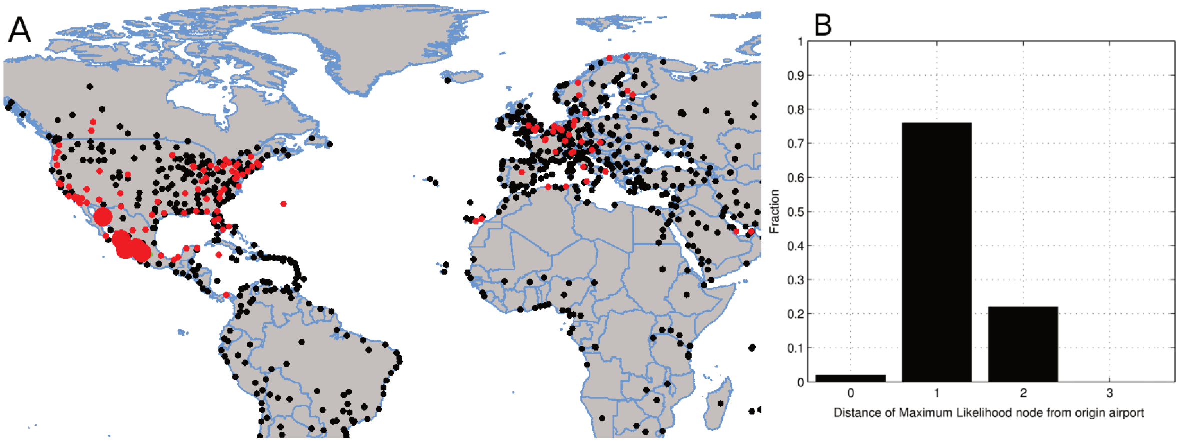


Figure 3.11: An example of source detection with different spreading model (SI spreading with p_{ij} from airport flux data, $T=10$) on a weighted network of the world airline transportation network from the Mexico City Juarez International airport, where red nodes represent infected airports, black ones non-infected and the big nodes represent top 5 maximum likelihood airports for this realization, which are all inside the Mexico state. Embedded histogram of maximum likelihood node topological distance to the true origin in 100 independent experiments is on the right.

Chapter 4

Backward in time - multiple source epidemic recognition

4.1 Problem formulation

Let us recall the definition of the random vector $\vec{R} = (R(1), R(2), ..., R(N))$, that indicates which nodes got infected prior to some temporal threshold T (random variable or constant) with the SIR stochastic process (p,q) on network G. The random variable $R(i)$ is a Bernoulli random variable, which assigns the value of 1 if node i got infected before time T from the start of the epidemic process and the value of 0 otherwise. Now, let us assume that we have observed one spatio-temporal epidemic realization $\vec{r_*}$ from the SIR process (p,q,T) and we want to infer whether it is more likely that the realization $\vec{r_*}$ comes from a single source set $S = \{\theta_1, \theta_2, ..., \theta_N\}$ or from some multiple source subset of S: $\left\{\left\{\theta_i, \theta_j\right\}, ..., \left\{\theta_i, \theta_j, \theta_k\right\}, ..., \left\{\theta_i, ..., \theta_l\right\}\right\}$. The number of potential multiple sources is equal to the sum: $\binom{N}{2} + \binom{N}{3} + ... + \binom{N}{N-1} + \binom{N}{N}$, which has in total $O(2^N)$ parts. We formulate two interesting research questions:

1. What are the differences in properties of epidemic propagation from multiple sources compared to single source epidemics ?
2. Can we make a posteriori estimate that some propagation is classified as a single source or as a multiple source ?

First, we shall concentrate only on single and 2-source epidemic processes and later deal with more complicated situations. Note, that the multiple source epidemic spreading could be an indicator of a terrorist attack as it is unlikely that viruses mutates simultaneously at different sources.

4.2 Statistical properties of 2-source epidemic processes

For a specific pair of nodes (i, j) from the contact network $G(V,E)$ and a specific SIR process (p,q) observe the following quantities:

- Probability distribution $P(X = k|\Theta = i)$,
- Probability distribution $P(X = k|\Theta = j)$,
- Probability distribution $P(X = k|\Theta = (i, j))$,

where X denotes the discrete random variable which measures the number of nodes that got infected during the epidemic SIR process (p,q) with initially infected nodes Θ. Alternatively, X denotes the number of nodes in the recovered compartment at the end of the epidemic process because all nodes that are in the recovered compartment at the end of epidemic also were infected during the epidemic process. We are interested in determining the difference in epidemic outcomes between probability distribution from a single node: $P(X = k|\Theta = i)$, $P(X = k|\Theta = j)$ and distribution from 2-sources $P(X = k|\Theta = (i, j))$. The difference between probability distributions can be measured with the symmetrised Kullback-Leibler divergence but for simplicity we used first moments of the random variable X: $E(X|\Theta = (i, j))$, $E(X|\Theta = i)$ and $E(X|\Theta = j)$. We define two quantities Δ_i^{ij} and Δ_j^{ij}:

$$\Delta_i^{ij} = \frac{E(X|\Theta = (i,j)) - E(X|\Theta = i)}{E(X|\Theta = (i,j))}, \Delta_j^{ij} = \frac{E(X|\Theta = (i,j)) - E(X|\Theta = j)}{E(X|\Theta = (i,j))},$$

where Δ_i^{ij} tell us what is the relative expected difference in number of infected nodes when epidemic process starts from (i, j) and i.

Therefore, for each pair of nodes (i, j) we can define the quantity Δ_{ij} as the average value of $(\Delta_i^{ij}, \Delta_j^{ij})$: $\Delta_{ij} = (\Delta_i^{ij} + \Delta_j^{ij})/2$. In order to get a better insight into Δ_{ij} quantity, we sample pairs (i, j) from network G with different geodesic distances between the nodes $d(i, j)$.

In figure 4.1, we show quantity Δ_{ij} as a function of distance $d(i, j)$ between pairs of nodes (i, j), which were sampled from the network of condensed matter collaborations [53] (cond-mat-2003) with pivot sampling by distance procedure (see the next subsection). Pivot sampling by distance procedure ensures that we have equal number of pairs with specific distance d. In figure 4.1, we can see that the mean value of Δ_{ij} is increasing with distance d, which confirms our hypothesis that the distance between pairs is one of the crucial factors that influence Δ_{ij}. When two nodes are close to each other, it is hard to distinguish 2-source from single source epidemic. But, in other limit the pair of nodes are far away from each other and it is much easier to distinguish 2-source from single source epidemic.

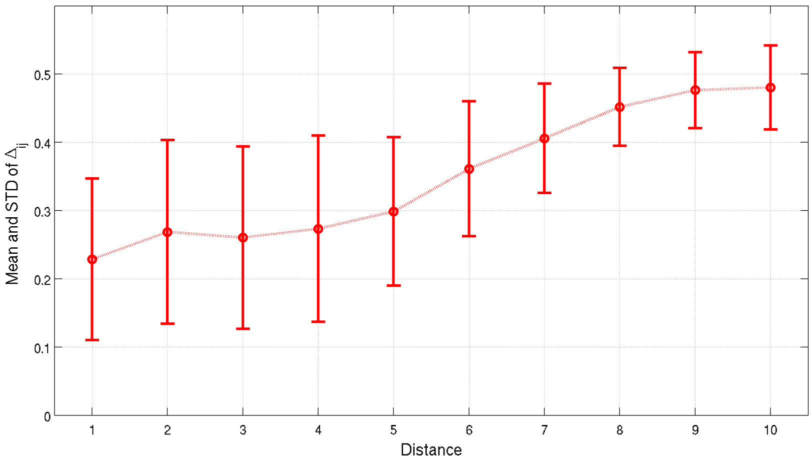


Figure 4.1: The quantity Δ_{ij} as a function of distance $d(i,j)$ for SIR process $(p=0.2, q=0.8)$ with $n=300$ simulations and "pivot sampling by distance" procedure with $m=100$ node pairs samples per distance. Simulations were performed on the network of co-authorships (cond-mat-2003) with diameter 14.

Interference and process decoupling

The connection of the expectations of the following probability distributions: $P(X=k|\Theta=i)$, $P(X=k|\Theta=j)$ and $P(X=k|\Theta=(i,j))$ is given with the following mathematical relation:

$$E[X|(i,j)] = E(X|i) + E(X|j) - \sum_{k\in V} P(\{i\to k\}\bigcap\{j\to k\}\,|(i,j)),$$

where

- $E[X|(i,j)]$ denotes the expected number of infected nodes when (i,j) are initial infected nodes,
- $E(X|i)$ denotes the expected number of infected nodes when i is initial infected node,
- $E(X|j)$ denotes the expected number of infected nodes when j is initial infected node,
- $P(\{i\to k\}\bigcap\{j\to k\}\,|(i,j)$ denotes the probability that node k was infected by node i and node j when (i,j) are initial infected nodes.

The last term we call the interference or overlap $I(i,j)$ between node i and j because it tell us the expected number of nodes in network that were infected by node i and node j.

$$I(i,j) := \sum_{k\in V} P(\{i\to k\}\bigcap\{j\to k\}\,|(i,j))$$

Now, the relation is written in very simple and intuitive manner:

$$E[X|(i,j)] = E(X|i) + E(X|j) - I(i,j).$$

If the ratio of the interference and expected number of infected nodes from (i,j) pair is smaller than some threshold T, then the epidemic process starting from (i,j) pair can practically be decoupled:

$$\frac{I(i,j)}{E[X|(i,j)]} < T \implies \hat{E}[X|(i,j)] \approx E(X|i) + E(X|j).$$

Now, we can also express the quantity Δ_{ij} via an interference $I(i,j)$:

$$\Delta_{ij} = \frac{E[X|(i,j)] - I(i,j)}{2E[X|(i,j)]} = \frac{1}{2}\left(1 - \frac{I(i,j)}{E[X|(i,j)]}\right).$$

From this relation, we can conclude that when the interference between nodes i and j is small the difference between 2-source and single source epidemic outcomes Δ_{ij} is large. On contrary, the difference between epidemic outcomes Δ_{ij} is small when the interference $I(i,j)$ is large.

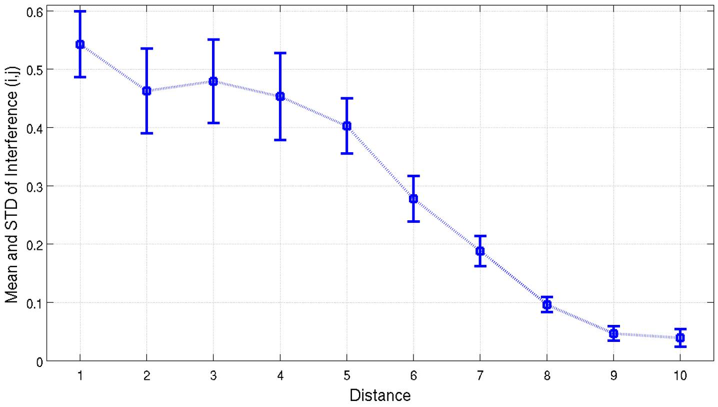


Figure 4.2: Mean and standard deviation of normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ for SIR process $(p = 0.2, q = 0.8)$ with $n = 300$ simulations and "pivot sampling by distance" procedure $m = 100$ pairs per distance. Simulations were performed on the network of co-authorships (cond-mat-2003) with diameter 14.

In figure 4.2, we can see the normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ between nodes. We can also see the errors we are making if we decouple the epidemic process starting from two nodes. If we decouple the epidemic process from two nodes (i,j) on distance 8, we are making an error of 10% of $E[X|(i,j)]$.

In Figure 4.3, we can see normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ between nodes and different values of SIR process parameters (p,q). High values of standard deviation for small values of parameter p could indicate bimodal behaviour [23] of the probability distribution of the number of infected nodes.

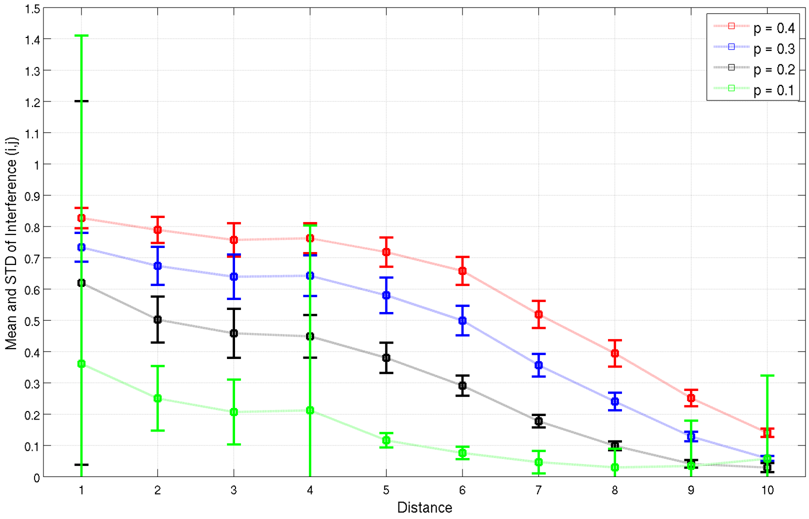


Figure 4.3: Mean and standard deviation of normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ for SIR process $q = 0.8$ and $p = (0.1, 0.2, 0.3, 0.4)$ with $n = 300$ simulations and "pivot sampling by distance" procedure $m = 500$ pairs per distance. Simulations were performed on the network of co-authorships (cond-mat-2003)

Pivot sampling by distance procedure

Let us denote the geodesic distance between nodes i and j in a network $G = (V, E)$ as a function $d(i, j)$. Let us define the distance pair sequence of length k to be sequence of pairs: $s = ((u_1, v_1), (u_2, v_2), ..., (u_k, v_k))$ such that $d(u_1, v_1) = 1$, $d(u_2, v_2) = 2$, ..., $d(u_k, v_k) = k$.
Sampling procedure satisfies the following constraints:

- all pairs in distance pair sequence are mutually independent
- distance pair sequences are mutually independent

We generate m independent pair distance sequences $\{s_i\}$, first by sampling pivot random node u from the network G and then we sample random node v from k-th neighbourhood of pivot node u such that: $d(u, v) = k$ and append pair (u, v) to sequence s_i.

Random pair sampling procedure

The previous results were created by using "pivot sampling procedure by distance". But, in order to be sure that our results are not artefacts of sampling procedure we have done another set of simulations with different sampling procedure.

Here, we sample the pairs of nodes from network G completely random. This sampling procedure is unbiased and we get some Gaussian-like distribution of distances between sampled pairs (see Figure 4.4).

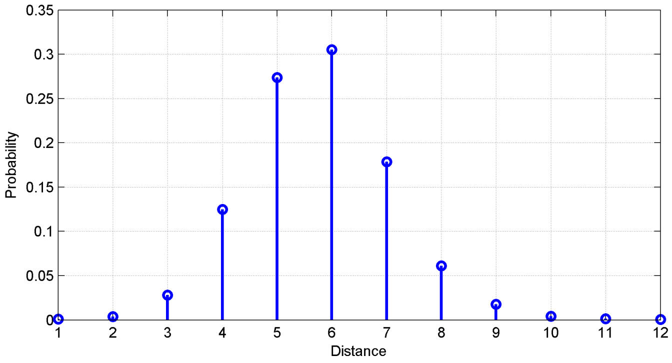


Figure 4.4: Probability distribution of distances among 10 000 pairs from the network of co-authorships (cont-mat-2003) when random pair sampling procedure was done.

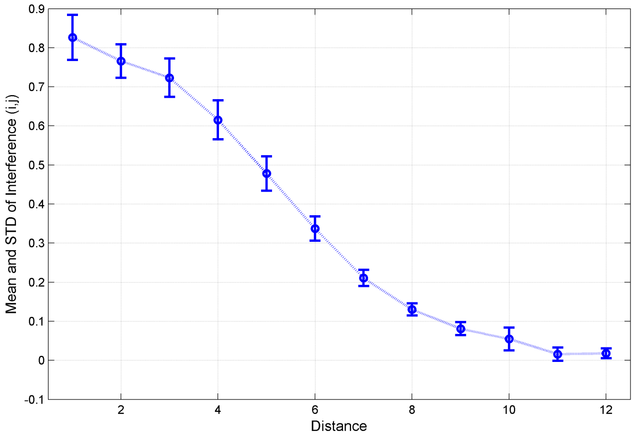


Figure 4.5: Normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ for SIR process $(p = 0.2, q = 0.7)$ with $n = 300$ simulations and "random sampling procedure" on 10 000 pairs. Simulations were performed on the network of co-authorships (cont-mat-2003)

From the Figure 4.5, we can see that the shape of the normalized interference $\frac{I(i,j)}{E[X|(i,j)]}$ as a function of distance $d(i,j)$ with "random sampling procedure" is similar to the shape of normalized interference function with "pivot sampling by distance procedure" (Figure 4.2).

4.3 Outlier detection method

In this section, we describe our statistical method for classifying the multiple-source realizations versus single source realizations with an outlier detection approach. Let us assume that we need to classify the observed realization $\vec{r}_*$. Our null hypothesis H_0 is: the observed realization $\vec{r}_*$ comes from one of the following single sources $S = \{\theta_1, \theta_2, ..., \theta_N\}$. Alternative hypothesis H_1 is: the observed realization $\vec{r}_*$ comes from some subset of the set S that has multiple sources:

$\{\{\theta_1,\theta_2\},...,\{\theta_i,\theta_j\},...,\{\theta_i,\theta_j,\theta_k\},...,\{\theta_i,\theta_j,\theta_k,...,\theta_l\}\}$. In order to eliminate the exponential combinatorial complexity of evaluating the likelihood of multiple sources, we will only evaluate likelihood of single sources w.r.t. observed realization $\vec{r}_*$ (accept or reject null hypothesis H_0). Multiple source epidemic realizations are considered as an outlier behaviour and single source epidemic realizations as a normal behaviour. Let us define the function $\varphi(\vec{r}_1,\vec{r}_2)$, which measures the similarity matching between realizations: $\vec{r}_1$ and $\vec{r}_2$. Now, we define new random variable $\varphi(\vec{r}_*,\vec{R}_\theta)$, which measures the φ similarity between the fixed realization $\vec{r}_*$ and random realization that comes from *SIR* process with the source θ. For each source θ we calculate the unbiased estimator of the cumulative distribution function of $\varphi(\vec{r}_*,\vec{R}_\theta)$ as the empirical distribution function $\hat{F}_{between}$ over n realizations $\vec{R}_{\theta,i}$ for specific θ and $\vec{r}_*$ as:

$$\hat{F}_{between}(\theta,\vec{r}_*,x) = \hat{P}(\varphi(\vec{r}_*,\vec{R}_\theta) \leq x) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[0,x\rangle}\left(\varphi(\vec{r}_*,\vec{R}_{\theta,i})\right)}{n},$$

where $\mathbf{1}_{[0,x\rangle}$ is a characteristic function defined like this:

$$\mathbf{1}_{[0,x\rangle}(y) = \begin{cases} 1 & : y \in [0,x\rangle, \\ 0 & : else, \end{cases}$$

Then, its probability density function is the following:

$$\frac{\partial}{\partial x}\hat{F}_{between}(\theta,\vec{r}_*,x) = \frac{1}{n}\sum_{i=1}^{n} \delta\left(x - \varphi(\vec{r}_*,\vec{R}_{\theta,i})\right),$$

where $\delta(x)$ is the Dirac delta function.

Now, we define a new random variable $\varphi(\vec{R}_\theta,\vec{R}_\theta)$, which measures the φ similarity within the realizations that comes from *SIR* process with the source θ. We also calculate the empirical distribution function of similarities within n realizations $\vec{r}_\theta(i)$ that come from single source θ:

$$\hat{F}_{within}(\theta,x) := \hat{P}(\varphi(\vec{R}_\theta,\vec{R}_\theta) \leq x) = \frac{\sum_i \sum_j \mathbf{1}_{[0,x\rangle}(\varphi(\vec{R}_{\theta,i},\vec{R}_{\theta,j}))}{\binom{n}{2}}.$$

Now, for fixed source θ we have empirical CDF functions of two random variables: $\varphi(\vec{r}_*,\vec{R}_\theta)$ and $\varphi(\vec{R}_\theta,\vec{R}_\theta)$ and we can calculate the Kolmogorov-Smirnov statistics between them:

$$KS_\theta = sup_x|\hat{F}_{between}(\theta,\vec{r}_*,x) - \hat{F}_{within}(\theta,x)|.$$

In the end, we have Kolmogorov-Smirnov statistics KS_θ for each single source from the set

$S = \{\theta_1, \theta_2, ..., \theta_k\}$ and we define new statistics which is a minimum of them:

$$KSM = min\{KS_{\theta_1}, KS_{\theta_2}, ..., KS_{\theta_k}\}.$$

Now, we make a statistical decision to reject the null hypothesis H_0 if

$$KSM \geq \alpha_{KS},$$

where α_{KS} is a statistical significance for two-sample Kolmogorov–Smirnov test for accepting realizations from a single source. If the null hypothesis has been rejected, we say that the observed realization $\vec{r}_*$ is an outlier w.r.t. probability distribution of *KSM* statistics.

The whole procedure to calculate *KSM* statistics for single realization $\vec{r}_*$ is explained in the following version of optimized KSM multiple source detection algorithm.

Algorithm 8 The KSM multiple source detection algorithm: $(G, p, q, \vec{r}_*, T, S, \alpha_{KS})$

```
Input: Network structure $G$, SIR process parameters $(p,q)$, $S = \{\theta_1, \theta_2, ..., \theta_k\}$ is a-priori set of possible sources $\theta_i$, observed realization $\vec{r}_*$ ending at some temporal threshold $T$, $\alpha_{KS}$ is a threshold for rejecting the single source hypothesis $H_0$
Downsample set $S$ to set of $l$ random probe nodes SP;
for each $\theta \in$ SP (Set of probe nodes) do
    for $i$ = 1 to $n$ (number of simulations) do
        - Run SIR simulation $(p,q,T)$ with $\theta$ and obtain epidemic realization $\vec{R}_{\theta,i}$;
        - Calculate and save $\varphi(\vec{R}_{\theta,i}, \vec{r}_*)$ ;
    end for
    Calculate $\hat{F}_{between}(\theta, \vec{r}_*, x)$:
        $\hat{F}_{between}(\theta, \vec{r}_*, x) = \hat{P}(\varphi(\vec{r}_*, \vec{R}_\theta) \leq x) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[0,x\rangle} \left(\varphi(\vec{r}_*, \vec{R}_{\theta,i})\right)}{n}$
    Calculate $\hat{F}_{within}(\theta, x)$ with $m = O(n)$ random samples:
        $\hat{F}_{within}(\theta, x) := \hat{P}(\varphi(\vec{R}_\theta, \vec{R}_\theta) \leq x) = \frac{\sum_{(s,t)} \mathbf{1}_{[0,x\rangle}(\varphi(\vec{R}_{\theta,s}, \vec{R}_{\theta,t}))}{m}.$
    Calculate Kolmogorov-Smirnov $KS_\theta$ statistics and save it
        $KS_\theta = sup_x |\hat{F}_{between}(\theta, \vec{r}_*, x) - \hat{F}_{within}(\theta, x)|;$
end for
Calculate $KSM$ statistics $KSM = min\{KS_{\theta_1}, KS_{\theta_2}, ..., KS_{\theta_l}\}$;
Output: Reject null hypothesis $H_0$ if $KSM \geq \alpha_{KS}$;
```

In order to calculate *KSM* statistics for single realization $\vec{r}_*$ with fixed pair (p,q), fixed network of size N with n simulations per each source we need to calculate in worst case $O(N * n)$

simulations and $O(N*n^2)$ similarity φ comparisons. For example, calculation of KSM statistics for single realizations on power-grid network (size $N \approx 5*10^3$) with $n \approx 5*10^2$ simulations per potential source takes us in a worst case approximately 10^6 independent SIR simulations on graph with N nodes and 10^9 similarity φ comparisons of realizations of size N.

Now, we explain the optimized version of KSM multiple source detection algorithm.
We have made multi-fold optimization procedures in order to execute experiments in a reasonable amount of time:

- Source downsampling: KSM statistic is estimated from l random "probe" nodes out of k potential sources, where $l \ll k$:

  $$\widehat{KSM} = min\{KS_{\theta 1}, KS_{\theta 2}, ..., KS_{\theta l}\}.$$

  Fisher–Tippett–Gnedenko theorem is describing the asymptotic distribution of $\widehat{KSM}$ statistics on i.i.d. samples of "probe" nodes. This statistic belongs to either Gumbel, Frechet or the Weibull family of extreme value distributions [61] (see Figure 4.6).
- Parallelization: by using MPI to share independent SIR simulations from l potential sources to the MPI process workers (reduction by number of MPI processes)
- Within similarities downsampling: we choose m independent pairs (s,t) of realizations from $\binom{n}{2}$ pairs of realizations from set $\left\{\vec{R}_{\theta,i}\right\}$ and estimate $F_{within}(x)$, where $m = O(n)$:

  $$\hat{F}_{within}(\theta, x) := \hat{P}(\varphi(\vec{R}_\theta, \vec{R}_\theta) \le x) = \frac{\sum_{(s,t)} \mathbf{1}_{[0,x\rangle}(\varphi(\vec{R}_{\theta,s}, \vec{R}_{\theta,t}))}{m}.$$

  Central limit theorem states that the pointwise, $\hat{F}_{within}(x)$ and $\hat{F}_{between}(x)$ have an asymptotic normal distribution. The rate at which this convergence happens is bounded by Berry–Esseen theorem. The approximation error of $\hat{F}_{between}$ and $\hat{F}_{within}$ is bounded by $O(n^{-1/2})$, where n is the number of simulations. We have made a small experiment, in which we demonstrate that the estimation error of $\hat{F}_{within}$ with $m = O(n)$ samples w.r.t $\hat{F}_{within}$ with $m = O(n^2)$ samples is small (see Figure 4.7).
- BitWise similarity φ calculation: realizations of size N are compressed to 64-bit unsigned integer array, reduction of realization array by a factor of 64 and calculation of $\overline{XNOR}$ and *Jaccard* with bitwise operations (XOR, NOT, AND) and bit count with Biran-Kernignan method [57].

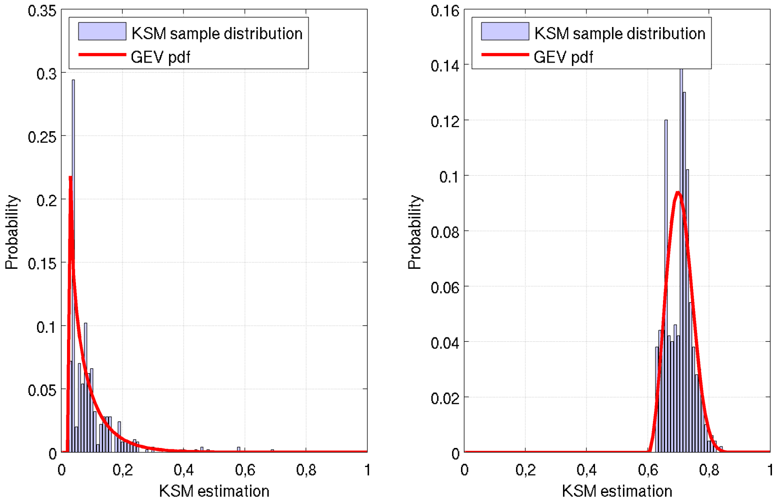


Figure 4.6: KSM estimation from 500 samples of size $l = 100$ and fitted with Maximum Likelihood Generalized Extreme Value Distribution on the power-grid network left for single source realization with ≈ 2000 infected nodes and right for 2-source realization with ≈ 2800 infected nodes.

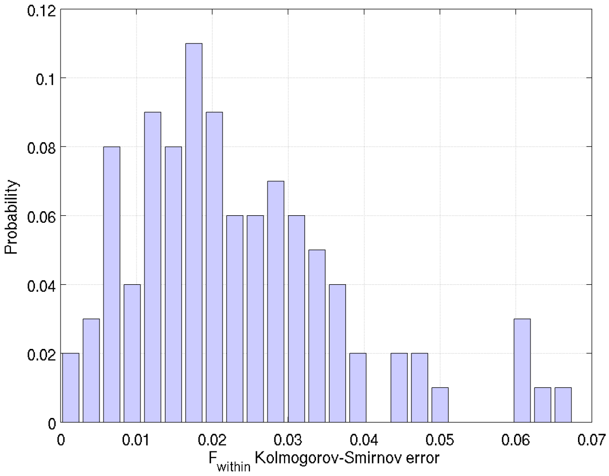


Figure 4.7: Kolmogorov-Smirnov error between $\hat{F^1}_{within}(x)$ with $m = O(n) = 10n$ samples and $\hat{F^2}_{within}(x)$ with $m = O(n^2) = n^2$ samples for power-grid network with random (p,q) parameters and random initial source node on 100 experiments with $T = 10$ and $n = 500$.

Non-trivial source multiplicity conditions

Now, we define trivial source multiplicity conditions:

- Condition 1: Realization $\vec{r}_*$ has more than one connected epidemic components.
- Condition 2: Diameter $D(\vec{r}_*)$ of a realization is greater than two times the duration of epidemic $D(\vec{r}_*) \geq 2T$.

Epidemic component is a subgraph which contains all nodes that were infected and corresponding edges of the original graph. If the realization satisfies any of two trivial condition then it has to have more than one initial source. Contrary, if the realization does not satisfy trivial conditions than both the single and multiple source are possible. In Figure 4.8, we can see two realizations of epidemics on regular grid that satisfies trivial (left) and non-trivial realization conditions (right).

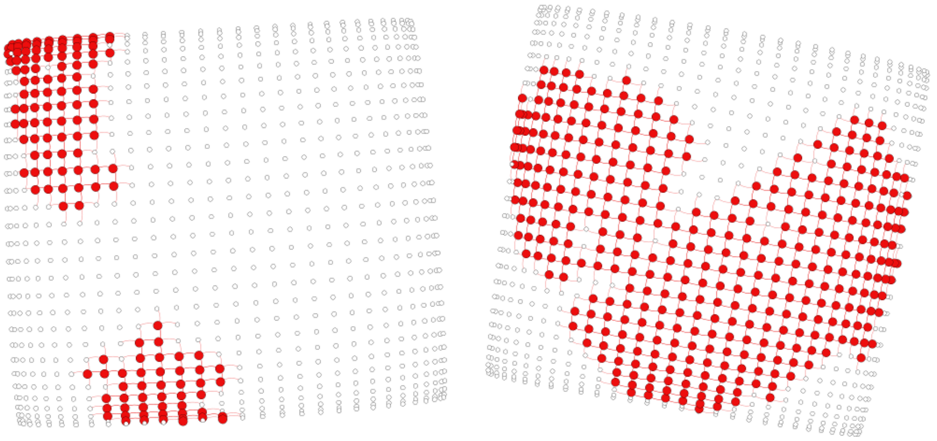
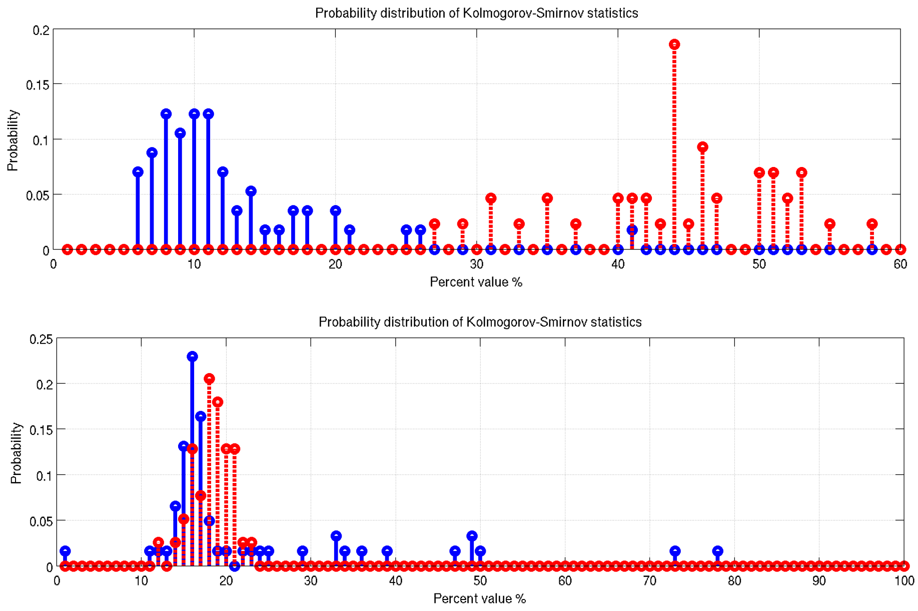


Figure 4.8: Two realizations of epidemics on regular grid that satisfies trivial multiplicity conditions. In left, we have two epidemic components and in the right the diameter of epidemic $D(\vec{r}_*) \geq 2T$ where $T = 10$.

4.4 Results

Now, we explain the experiments we have performed in order to demonstrate the performance of the optimized KSM multiple source detection algorithm. In Figure 4.9, we can see the distribution of *KSM* statistics for Erdös-Rényi network on 100 experiments. From this result, it is not clear when the detection of multiple-source epidemics is possible. In the following experiments, we try to get a clear picture what is influencing the possibility of multiple-source detection. We are studying the effects of network structure, SIR parameters (p,q) and temporal evolution threshold T on accuracy of detecting multiple-source realizations.

Figure 4.9: Distributions of KSM statistics for single and multiple-source non-trivial realizations (100 experiments) on Erdös-Rényi network (N = 5000, p = 0.001). In each experiment we classify realization that comes from single (blue) or multiple node (red). Up for SIR process $(p,q) = (0.2,0.2)$, temporal threshold $T = 5$ and down for SIR process $(p,q) = (0.3,0.4)$ and temporal threshold $T = 15$. Experiments for single (0.5 probability) vs (2,3,4,5) sources (0.5 probability)



All experiments are performed on realizations that satisfy the non-trivial realization condition 1 and the condition 2 is used as a baseline solution. In each experiment we observe realization and we calculate it's KSM statistics. Then, we are interested in the maximum classification accuracy over all thresholds α_{KS} for rejecting the single source hypothesis H_0. By constructing single source KSM distribution and multiple source KSM distribution, we can calculate maximal predictive accuracy over all α_{KS}. In figure 4.9, we can see single source KSM distribution denoted with blue color and multiple source KSM distribution denoted with red color.

In order to analyse the performance of multiple source detection we have used different similarity functions:

- Using KSM statistics on the similarity function which compares only the total number of infected nodes in realizations:

$$\varphi_I(\vec{r}_1, \vec{r}_2) = \frac{|V| - ||\vec{r}_1| - |\vec{r}_2||}{|V|}$$

  ,where $|V|$ denotes total number of nodes in graph and $|\vec{r}_1|$, $|\vec{r}_2|$ number of infected nodes in realizations $\vec{r}_1$ and $\vec{r}_2$.

- Using KSM statistics on the similarity function which compares the diameter of infected realizations:

  $$\varphi_D(\vec{r}_1, \vec{r}_2) = \frac{D(G) - (D(\vec{r}_1) - D(\vec{r}_2))}{D(G)}$$

  , where $D(G)$ is the diameter of the graph and $D(\vec{r}_1)$, $D(\vec{r}_2)$ are diameters of infected subgraphs in realizations $\vec{r}_1$ and $\vec{r}_2$.
- Using KSM statistics on the similarity function by comparing realizations with the XNOR similarity function:

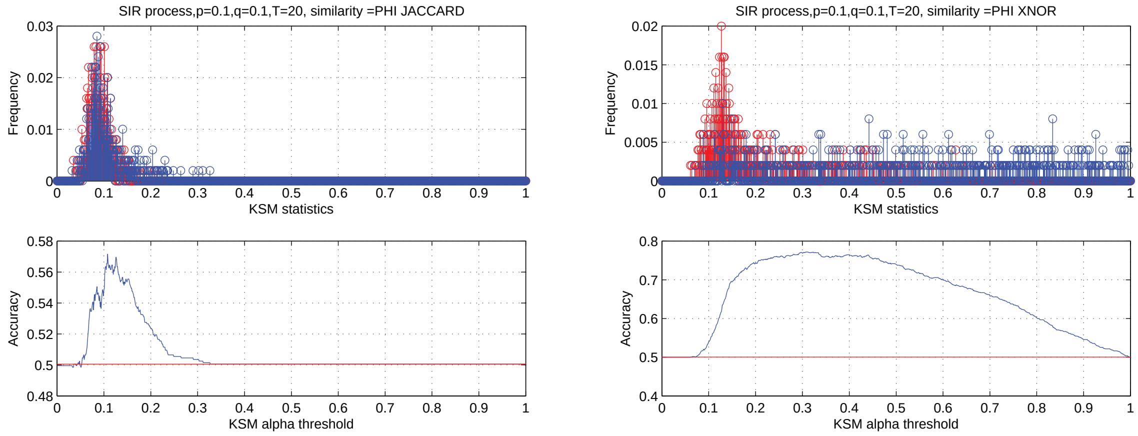
  $$\varphi_X(\vec{r}_1, \vec{r}_2) = \frac{\sum_{k \in V} \psi_{\oplus}(\vec{r}_1(k), \vec{r}_2(k))}{|V|}$$

  ,which count number of corresponding states in realizations normalized by total number of nodes.
- Using KSM statistics on the similarity function by comparing realizations with the Jaccard similarity function:

  $$\varphi_J(\vec{r}_1, \vec{r}_2) = \frac{\sum_{k \in V} \psi_{\wedge}(\vec{r}_1(k), \vec{r}_2(k))}{\sum_{k \in V} \psi_{\vee}(\vec{r}_1(k), \vec{r}_2(k))}$$

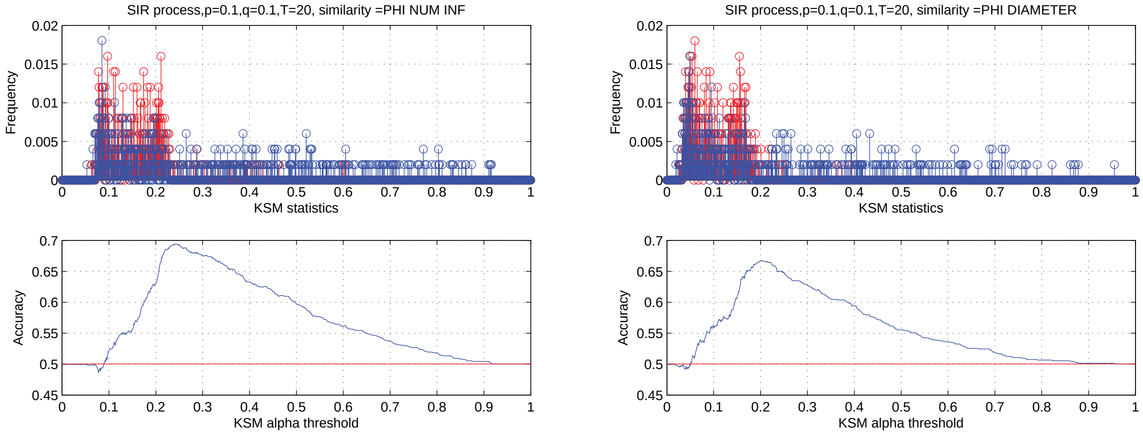
  ,which count number of corresponding states in realizations normalized by the size of the union of infected nodes in realizations.

In Figures 4.10, 4.11 and 4.12, we observe that the KSM statistics over φ_X function has stable results which outperform other similarity functions. From information theoretic perspective the φ_X and φ_J use data with more information content than φ_I function which only need the number of infected nodes in realizations. But still the φ_J can have the lower performance than φ_I. Possibly, this can be due to the normalization which is dependent on the size of the union of infected nodes in both realizations which can affect $\hat{F}_{within}(x)$ estimation. From computation complexity perspective the φ_D function has the highest computational complexity w.r.t. other measures as it need to calculate diameter of large number of realizations. In Figure 4.13 and 4.14, we observe the effects of different classes of networks on the predictability of multiple-sources.



(a) Jaccard similarity

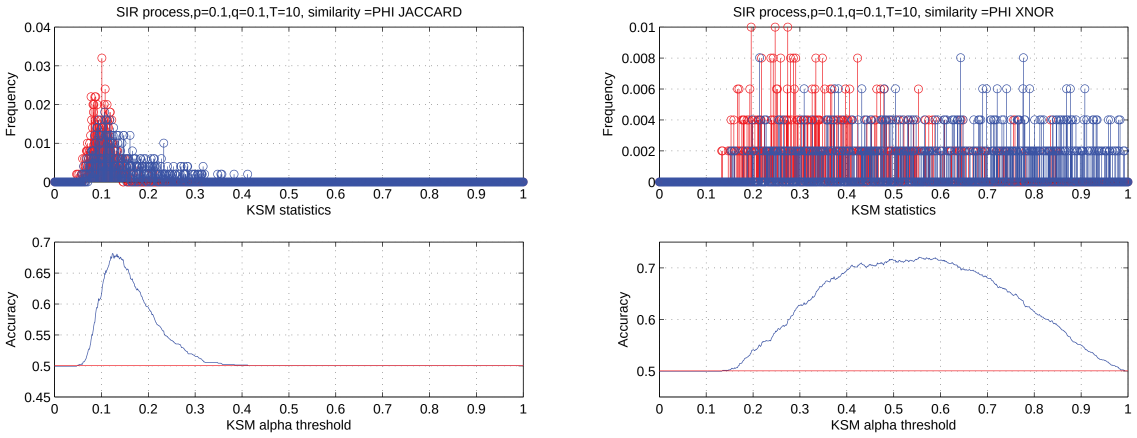
(b) XNOR similarity



(c) Similarity via number of infected nodes

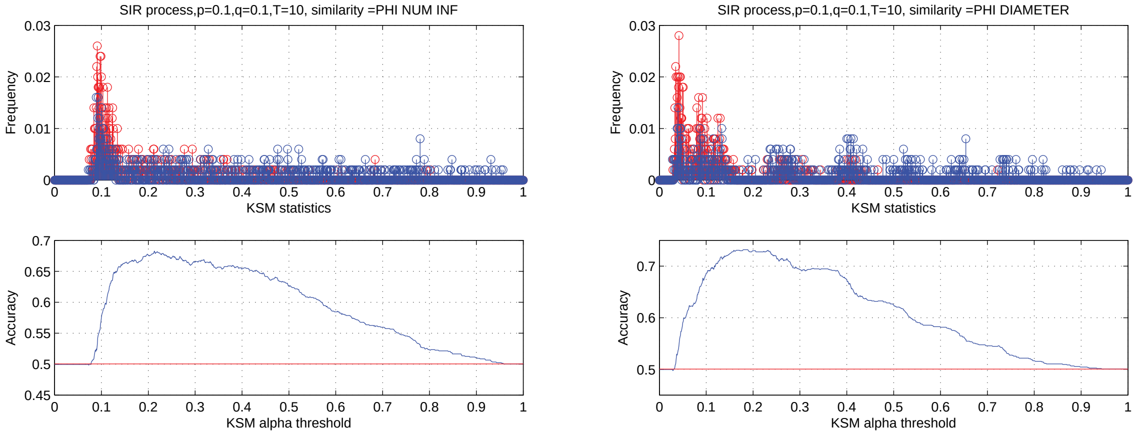
(d) Similarity via infected subgraph diameter

Figure 4.10: Accuracy of KSM statistics with different similarity functions on regular lattice grid for $T = 20$. By using the condition 2 to find trivial multiple source realizations when $T < \frac{D(\vec{r})}{2}$ we plot the baseline with red in accuracy plots.



(a) Jaccard similarity

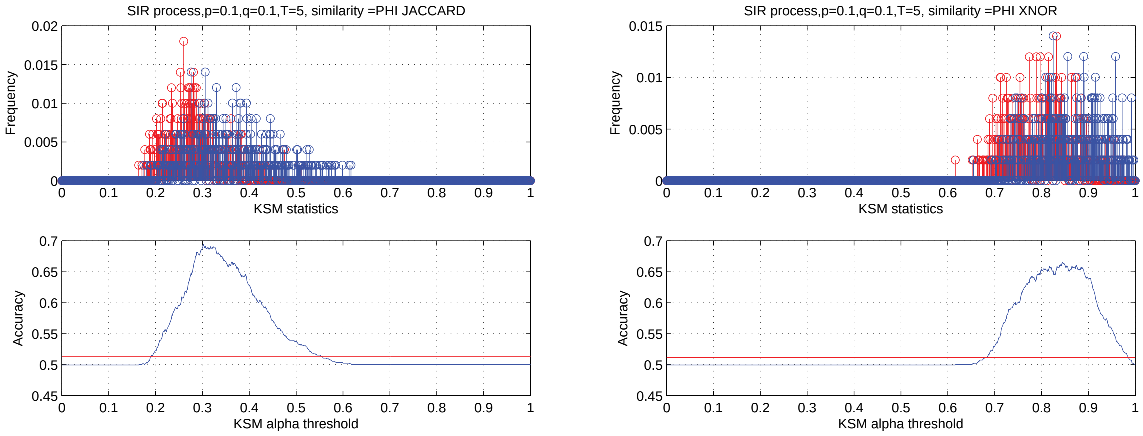
(b) XNOR similarity



(c) Similarity via number of infected nodes

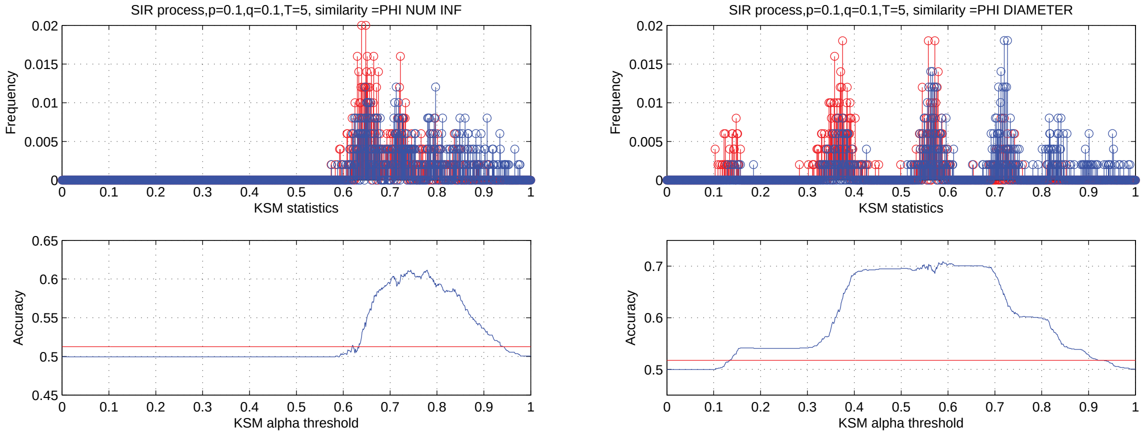
(d) Similarity via infected subgraph diameter

Figure 4.11: Accuracy of KSM statistics with different similarity functions on regular lattice grid for $T = 10$. By using the condition 2 to find trivial multiple source realizations when $T < \frac{D(\vec{r})}{2}$ we plot the baseline with red in accuracy plots.



(a) φ_J similarity

(b) φ_X similarity



(c) φ_I similarity

(d) φ_D similarity

Figure 4.12: Accuracy of KSM statistics with different similarity functions on regular lattice grid for $T = 5$. By using the condition 2 to find trivial multiple source realizations when $T < \frac{D(\vec{r})}{2}$ we plot the baseline with red in accuracy plots.

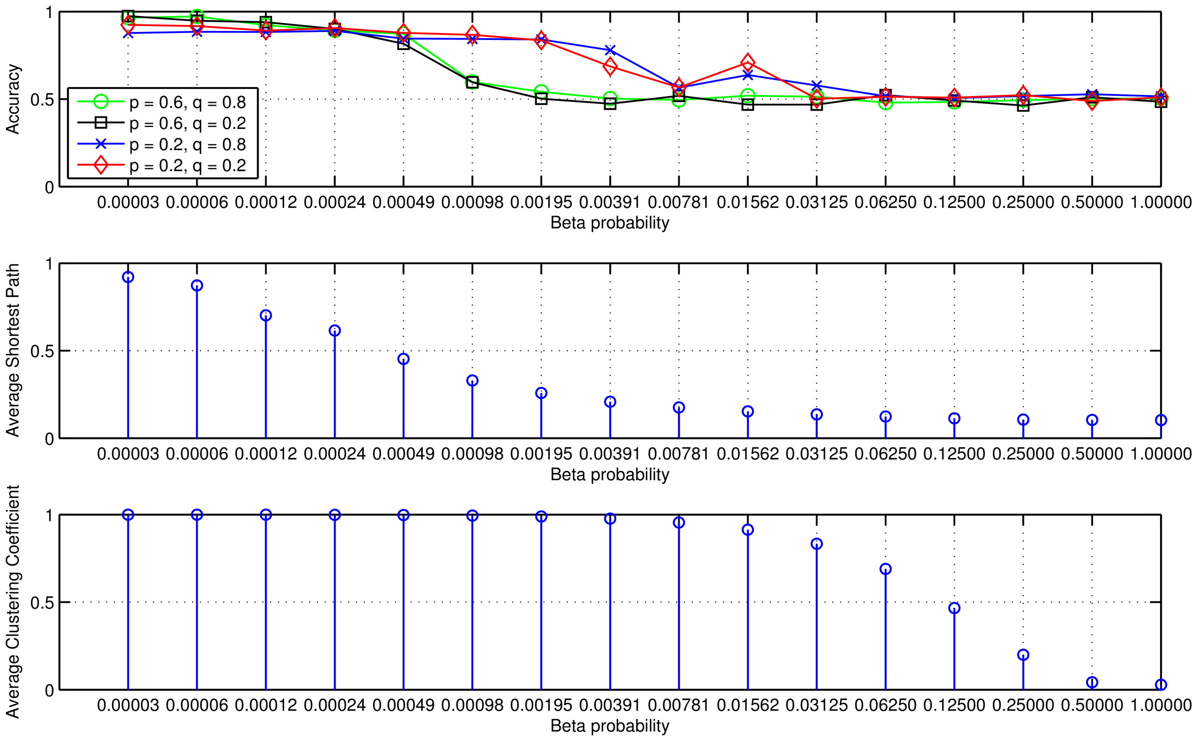


Figure 4.13: Maximal accuracy of rejecting the null hypothesis for different parameters of (p,q), $T = 10$ with 1000 experiments on classes of networks from regular lattice ($\beta = 0$) to random networks ($\beta = 1$) with Small-world networks in the middle. The average shortest path is normalized by average shortest path (≈ 120) in a regular lattice. The average clustering coefficient is normalized by average clustering coefficient (≈ 0.7) in a regular lattice. Experiments with single source (0.5 probability) vs two sources (0.5 probability). $\varphi_x(\vec{r}_1, \vec{r}_2)$ similarity measure was used.

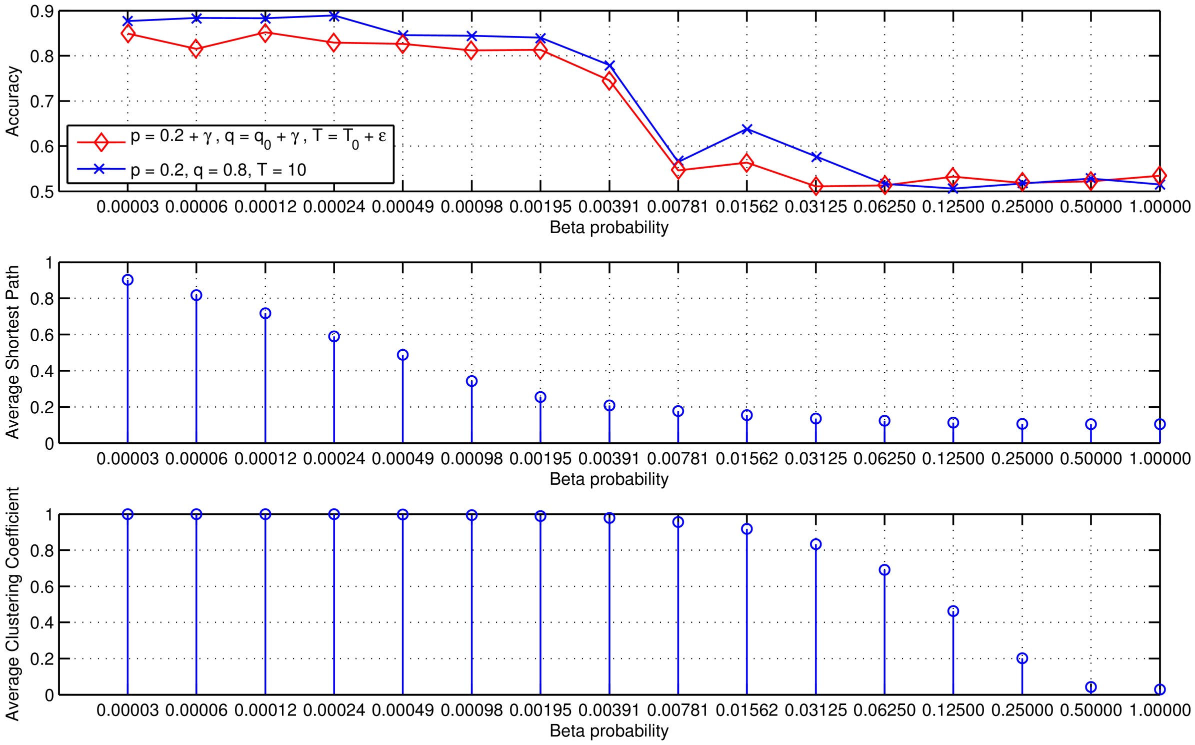


Figure 4.14: Maximal accuracy of rejecting the null hypothesis for different parameters of $p = p_0 + \gamma$, $q = q_0 + \gamma$, $T = T_0 + \varepsilon$, where $T_0 = 10$, $p_0 = 0.2$, $q_0 = 0.8$ with 1000 experiments on classes of networks from the regular lattice ($\beta = 0$) to random networks ($\beta = 1$) with Small-world networks in the middle. The average shortest path is normalized by average shortest path (≈ 120) in a regular lattice. Average clustering coefficient is normalized by average clustering coefficient (≈ 0.7) in a regular lattice. Experiments with single source (0.5 probability) vs two sources (0.5 probability). $\varphi_x(\vec{r}_1, \vec{r}_2)$ similarity measure was used. The ε noise was modelled with the geometric distribution with parameter 0.5 and γ noise variable as normally distributed random variables with parameters $(0, 0.05)$.

Chapter 5

Conclusion and discussion

In this thesis three objectives have been formulated, (i) the forward in time epidemic modelling, (ii) backward in time single source detection and (iii) backward in time multiple-source recognition.

The first objective has been achieved by constructing the FastSIR algorithm which estimates the final state for each node in an arbitrary network under the discrete SIR model. Although, different approximation techniques from statistical mechanics exists for epidemic modelling on complex networks, it is also very important to have both exact and fast statistical algorithm that captures all correlations in arbitrary networks and all correlations between node disease states. The FastSIR algorithm reduces the average case running time of the Naive SIR algorithm by approximately constant factor $1/q$ in the parametric space (p,q) and thus has the average case running time equal to total expected number of infected nodes times average node degree. One possible direction of future research is implementing different probability distributions of the number of infected nodes into the FastSIR algorithm, e.g. continuous time distributions or recovery time distributed not only with geometric, but with general negative binomial distribution.

The second objective has been achieved by constructing a set of statistical estimators (AUCDF, AvgTopk, Naive Bayes, Direct Monte Carlo and Soft Margin) for the epidemic source detection on arbitrary networks under the SIR discrete model. Various researchers have proposed different solutions to the problem of the epidemic source detection, which are based on a number of assumptions on contact network structures and spreading models. Detecting the source of an epidemic spreading under the stochastic SIR discrete model on arbitrary networks represents an extension of existing research methodologies, mainly focussed on the diffusion-like processes and specific networks. Furthermore, this statistical framework can be deployed for different kinds of stochastic compartment processes (ISS, SI, SIR, SEIR) on networks whose dynamical patterns can be described by probability distributions over similarities among realization vectors.

The AUCDF estimator, the AvgTopk estimator and the Naive Bayes estimator give a fast estimate about the ranking of the potential source candidates, but the Soft margin estimator can also reconstruct the source probability distribution and it generally gives better performance on benchmark cases with comparison to other estimators (including other state-of-the-art estimators like Jordan and DMP). Note, that the source probability distribution gives more information than just the ranking and also is important for estimating the detectability limits. Note, that the Soft margin estimator is controlled via one parameter a (a measure of a margin width) and a realization similarity function. When a goes to zero, the Soft margin estimator goes to the Direct Monte Carlo estimator, which is an unbiased estimator. Note, that this parameter a can be chosen automatically as the minimum value for which the source probability distributions have converged. The Soft margin estimator has been applied in a case study of sexual transmitted infection on an empirical temporal network of sexual contacts and in a diffusion of disease through the weighted world airport network. In future research about the epidemic source detection, one should try to construct an efficient Markov Chain Monte Carlo source detection estimator.

The third objective has been achieved by finding a multiple-source recognition algorithm, which can give a statical estimate whether the realization has one or more epidemic sources. Here the problem is mapped to an outlier detection problem by comparing two distinct distributions of similarities: (i) within the single-source to single-source realizations and (ii) between single-source realizations and observed realization. Then the Kolmogorov-Smirnov statistics between these distributions is used as a measure whether the observed realization is an outlier (multiple-sources) with respect to single-source realizations. Note, that in the third objective, an algorithm for recognition of multiple-source has been constructed and the problem of finding the locations of multiple sources on an arbitrary network is an important research question for future work.

Appendix A

Fundamentals of complex network theory

A network is a set of nodes interconnected by a set of links. Network, nodes and links are also called graph, vertices and edges, respectively, in the graph theory literature [62]. Here, we will mostly use the terms: network, vertices and edges. We will also study only simple networks - a network that contains no multiedge nor a self-edge.

Mathematics of networks

An adjacency matrix is one way of representing a network structure. The adjacency matrix A of a simple network is the matrix which contains non-zero element A_{ij} if there exists an edge between vertices i and j. The non-zero element A_{ij} for an unweighted network is 1 and for a weighted network is an arbitrary number called a weight. The degree of a vertex in undirected network is the number of edges connected to it. In directed networks edges have property of direction, therefore an adjacency matrix contains a non-zero element A_{ij} only if there exists an edge from i to j. Note that the adjacency matrix of a directed graph is asymmetric in general and symmetric for the undirected networks. In the directed networks, we define the in-degree and the out-degree as the number of ingoing and outgoing edges, respectively.

It is sometimes more convenient to convert a directed network to an undirected network. One simple approach is to make all the edges symmetric. The second approach is to construct co-citation network C. The co-citation of two vertices i and j are the number of other vertices that both have outgoing edges to i and j. The adjacency matrix of the co-citation network $C = AA^T$ is calculated from the adjacency matrix of the directed network and putting all the diagonal elements to zero.

A tree is a connected, undirected network without cycles. A directed network without any cycles is called an acyclic directed network. A citation network is an example of an acyclic directed network. For every acyclic directed network there exists labeling of the vertices such that the adjacency matrix is strictly upper triangular. The eigenvalues of the adjacency matrix are zeros if and only if the network is an acyclic directed network.

In a bipartite network (e.g. movie recommendation network) there exist two types of vertices (users - type$_1$ and movies - type$_2$) and edges which connect only vertices of different types. The structure of bipartite network is represented with a rectangular incidence matrix. Incidence matrix B has non-zero elements B_{ij} if a vertex of type$_1$ is connected to a vertex of type$_2$. Adjacency matrices P_1 and P_2 of one-mode projections of the bipartite network to a type$_1$ or a type$_2$ network are calculated from the incidence matrix $P_1 = BB^T$, $P_2 = B^T B$ and setting all the diagonal elements to zero.

Another very important matrix that is used to represent a network structure is the Laplacian matrix L of the network. The Laplacian matrix is calculated as $L = D - A$, where A is the adjacency matrix and D is the diagonal matrix with the degrees k_i of nodes on the diagonal. The Laplacian matrix is a symmetric matrix (for undirected networks), therefore its eigenvalues are real. Furthermore, the Laplacian matrix is a positive semi-definite matrix because it can be decomposed as $L = B^T B$, where B is the edge incidence matrix. Note that the Laplacian matrix is a singular matrix because the Laplacian matrix always has at least one zero eigenvalue with the corresponding eigenvector 1 .

$$\sum_m L_{im} \times \vec{1} = \sum_m (D_{im} - A_{im}) \times \vec{1} = \sum_m (\delta_{im} k_m - A_{im}) \times \vec{1}$$

$$= k_i - \sum_m A_{im} = k_i - k_i = 0$$

A network with n components has n zero eigenvalues and n different corresponding eigenvectors $\vec{e}_i$. The eigenvector $\vec{e}_i$ of the i-th component contains ones in all the places k where node k is in the i-th component and zeros elsewhere. The number of zero eigenvalues, is equal to the number of components in the network. Therefore, if the network contains only one component, the second eigenvalue (the algebraic connectivity) is positive. The Perron-Frobenius theorem tells us that each real non-negative irreducible matrix (undirected fully connected graph) has unique largest eigenvalue whose corresponding eigenvector contains non-negative values.

Measures and metrics

Although, a degree centrality is the most intuitive measure, it does not capture most influential vertices in the network. The eigenvector centrality [63] is based on a simple concept, a vertex is more important if it has more important neighbours. If we denote x_i as the importance of vertex i, then the eigenvector centrality can be calculated with the following expression: $x_i = \sum_k A_{ik} x_k$. This expression assumes that we already know importances of the neighbours and so on recursively. We can calculate importances iteratively like this $\vec{x}(t) = A^t \vec{x}(0)$ by setting a vector $\vec{x}(0)$ to some arbitrary value. The initial vector can be expressed as: $\vec{x}(0) = \sum_j a_j \vec{e}_j$, the linear combination of the eigenvectors of the adjacency matrix. Finally, we can calculate the eigenvector centrality for all the vertices in the network by using only leading eigenvector $\vec{e}_1$ and eigenvalue λ_1 of the adjacency matrix A.

$$x(t) = A^t \sum_k a_k \vec{e}_k = \sum_k a_k \lambda_k^t \vec{e}_k = \lambda_1^t \sum_k a_k \frac{\lambda_k^t}{\lambda_1^t} \vec{e}_k$$

In the limit of the time, we express the eigenvector centralities with only one leading eigenvector ($A\vec{e_1} = \lambda_1 \vec{e_1}$), which elements are all non-negative. Nevertheless, the eigenvector centrality fails as an importance measure on the directed networks, so we introduce the PageRank centrality.

The PageRank citePageRank centrality measure of the particular vertex is proportional to the neighbours PageRank centrality divided by their out-degree. This can be written in matrix terms like this:

$$\vec{x} = \alpha A D^{-1} \vec{x} + \beta \vec{1},$$

where A is the adjacency matrix, D the diagonal matrix with the elements $D_{ii} = max(k_i^{out}, 1)$ down the diagonal. Note that the PageRank centrality on undirected networks is reduced to the vertex degree.

Kleinberg [64] introduced two different types of the centrality importance for the vertices in the directed networks. Each vertex in the network can have a hub centrality (contain information about the best authorities) and an authority centrality (contain useful information). The authority centrality x_i of the vertex v_i in the network is proportional to the sum of the hub centralities y_j of the vertices that have out-going edge to the i-th vertex v_i:

$$x_i = \alpha \sum_j A_{ij} y_j.$$

The hub centrality y_i of a vertex v_i in the network is proportional to the sum of the authority centralities x_j of the vertices that have in-going edge from the i-th vertex v_i:

$$y_i = \beta \sum_j A_{ji} x_j.$$

In matrix terms, we can write $\vec{x} = \alpha A \vec{y}$ and $\vec{y} = \beta A^T \vec{x}$. From there we can write $AA^T \vec{x} = (\alpha\beta)^{-1} \vec{x}$ and $A^T A \vec{y} = (\alpha\beta)^{-1} \vec{y}$. Therefore, we conclude that the hub and the authority centrality are the leading eigenvectors from matrices AA^T and $A^T A$ from the same eigenvalue $(\alpha\beta)^{-1}$, respectively. This procedure for computing hubs and authorities centrality is used in the HITS algorithm.

A path in the network is defined as an arbitrary sequence of vertices. Number of paths, between the vertices i and j, with the given length k, can be computed from the adjacency matrix: A^k_{ij}. The number of cycles of length k in the network can be computed as a sum over all vertices: $\sum_m A^k_{mm}$, which is equal to the trace of the matrix A^k, which is equal to the sum of eigenvalues of the matrix A^k. A geodesic path is the shortest path between two vertices. Let us denote d_{ij} as the length of the geodesic path from a vertex i to a vertex j. The closeness centrality C_i of vertex v_i is the harmonic mean between the distances of geodesic paths from the vertex v_i to all others.

$$C_i = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}}.$$

We denote σ_{st} as the number of geodesic paths between pairs of vertices v_s and v_t and the $\sigma_{st}(v_i)$ as the number of the geodesic paths σ_{st} which pass through the vertex v_i. Then the betweenness centrality is defined as:

$$C(v_i) = \sum_{st} \frac{\sigma_{st}(v_i)}{\sigma_{st}}.$$

The degree distribution $P(k)$ defines the probability of choosing a vertex with the degree k by uniform sampling from the set of all vertices. The n-th moment of $P(k)$ is calculated as:

$$\langle k^n \rangle = \sum_k k^n P(k).$$

We can also define the average degree of the nearest neighbours of the nodes with the degree k as:

$$k_{nn}(k) = \sum_{k'} k' P(k'|k).$$

In uncorrelated networks, $k_{nn}(k)$ is independent of k. The correlated networks are called assortative if $k_{nn}(k)$ is an increasing function of k. If $k_{nn}(k)$ is a decreasing function of k then the network is disassortative [65]. The local clustering coefficient C_i is defined as the ratio of the number of edges e_i between first neighbours of v_i and the number of all possible edges between them.

$$C_i = \frac{2e_i}{k_i(k_i-1)}$$

Topology of real networks

Most of real networks in information, social and biological systems are characterized by the similar topological properties: small average path length, high clustering coefficients, fat tailed scale-free degree distributions, degree correlations and presence of communities.

If the average shortest path length in the network depends logarithmically on the network size, the network is considered to have the small-world property. Most real networks have power law degree distribution $P(k) = Ak^{-\gamma}$, where γ is in the range $2<\gamma<3$. The networks with the power law distribution are called scale-free networks (have the same functional form

Table A.1: Topology characteristics of real networks: size-N, average degree-$\langle$k$\rangle$, average path length-L, average clustering coefficient-C, exponent of power-law distribution-γ, correlation of degrees between connected nodes-ν

Network	N	$\langle$k$\rangle$	L	C	γ	ν
Internet Autonomous systems [66]	11,174	4.19	3.62	0.24	2.38	<0
WWW [67]	2×10^8	7.5	16	0.11	2.1/2.7	-
Protein [68]	2,115	6.8	2.12	0.07	2.4	<0
Metabolic [69]	778	3.2	7.4	0.7	2.2/2.1	<0
Mathematical co-authorship [70]	57,516	5.0	8.46	0.15	2.47	>0
Actors [11]	225,226	61	3.65	0.79	2.3	>0

at the different scales) [12],[13]. Finite-size networks exhibit cutoffs in the fat-tailed degree distributions [71]. Distributions whose tails are not exponentially bounded are called the fat-tailed or the heavy-tailed distributions.

Modeling network structure

Modeling global network structure

Random graphs were first studied by Erdös and Rényi in 1959. Their first model generated Erdös and Rényi random graphs [72] with N vertices and K edges from an entire statistical ensemble of all possible realizations. Later, another model for the ER random graphs was presented, which generates a random graph of N vertices where the probability of an edge occurrence is p. The graphs with k edges will appear in the ensemble with the probability $p^k(1-p)^{N(N-1)/2-k}$ [72]. The structural properties of the ER random graphs exhibit a phase transition at the critical probability $p_c = \frac{1}{N}$. When $p<p_c$ the graph almost surely has no components of size greater than $O(ln(N))$. Above that critical probability the random graph has a component of $O(N)$. The degree distribution follows the binomial distribution:

$$P(K=k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}.$$

For a large value of N and fixed $\langle k \rangle$ the degree distribution can be represented by the Poisson distribution:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}.$$

Although the ER random graphs are well mathematically explained they do not reproduce the topological properties of real networks.

The configuration model allows to sample random graphs from the ensemble with the arbitrary degree distribution $P(k)$ and N vertices. A generalized random graph is constructed by assigning a k_i half-edges to the vertex v_i and wiring half-edges with uniform probability.

The Watts and Strogatz model generates small-world networks with a high clustering coefficient [11]. The model starts from the N vertex ring where each vertex is connected to its $2m$ nearest neighbours. Then the process of rewiring starts and each edge is rewired to the randomly chosen vertex with the probability p. The regular lattice occurs when p equals to zero and a random graph occurs when p equals to one. For intermediate values of p, small-world networks with a high clustering coefficient occur. The small rewiring procedure has a huge nonlinear effect on decreasing an average shortest path L and a linear effect on decreasing a clustering coefficient.

The Barabási-Albert model (BA) is the model of evolving a scale-free network, which uses a preferential attachment [13] property. Starting from the m_0 isolated vertices, at each time step new vertices with m edges are added to the network ($m < m_0$). The new vertex will create an edge to the existing node v_i with the probability proportional to its degree k_i. The BA model produces the power-law distribution $P(k) \sim k^{-3}$ in the limit of time. The average distance increases logarithmically with the size of the network. The clustering coefficient vanishes with the system size slower than in ER random graphs, but still different from small-world models where C is a constant. Various authors have proposed modifications and generalizations of the standard BA model in order for it to become more realistic.

Although various network models have been constructed, they fail to reproduce several properties like: the scree plot (eigenvalues in descending order), the densification law or the shrinking diameter property. The eigenvalues versus their corresponding rank of the adjacency matrix are represented by the scree plot, this plot also obeys a power-law. The densification power law tells us that the relation between the number of edges over time $E(t)$ and the number of vertices over time $V(t)$ in the evolving network is: $E(t) = V(t)^a$ (the densification exponent a is greater than 1) [73]. The effective diameter of the network tends to shrink in an evolving network [73]. A Kronecker graph K_1^k is defined by a k recursive Kronecker product of an initiator graph K_1.

$$K_1^k = K_k = \underbrace{K_1 \oplus K_1 \oplus \ldots \oplus K_1}_{k} = K_{k-1} \oplus K_1.$$

The Kronecker graphs have a multinomial distribution for in and out degrees, eigenvalues, components of leading eigenvector and follow the densification law [74]. For some choice of the initiator K_1, the multinomial distribution behaves like a power-law distribution. Stochastic Kronecker Graphs have also been introduced [74], where values of K_k are probabilities of edges.

The Kronecker graphs can model the real networks by tuning the parameters in the initiator matrix K_1. For the given graph G with the N_1^k vertices and the initiator matrix K_1 ($N_1 \times N_1$) one can generate the Kronecker graph K_k with the N_1^k vertices and calculate the likelihood between the graph G and the Kronecker graph K_k like this:

$$P(G|K_1) = \prod_{(u,v)\in G} K_k(u,v) \prod_{(u,v)\notin G} (1 - K_k(u,v)).$$

Calculating the likelihood by this approach has two problems. The first problem is matching the corresponding vertices between the adjacency matrix of G and the adjacency matrix of the Kronecker graph K_k (factorial problem). The second problem is a complexity of calculating the likelihood when the vertices have been matched $O(N^2)$. By using a Markov Chain Monte Carlo method (the Metropolis sampling algorithm) for a vertex matching and the Taylor approximation of likelihood, calculations of the likelihood can be done in linear time $O(E)$. But, we want to find the initiator matrix K_1 such that has the maximum likelihood $P(G|K_1)$. For simplicity, we will denote the initiator matrix K_1 as Θ and one possible matching of vertices as σ. The log-likelihood can then be written as:

$$l(\Theta) = logP(G|\theta) = log\sum_{\sigma} P(G|\Theta,\sigma)P(\sigma,\Theta).$$

To maximize the likelihood $P(G|\Theta)$, the gradient method can be employed:

$$\hat{\Theta}_{t+1} = \hat{\Theta}_t + \lambda \frac{\partial l(\Theta)}{\partial \Theta}.$$

where the gradient is:

$$\frac{\partial l(\Theta)}{\partial \Theta} = \sum_{\sigma} \frac{\partial logP(G|\sigma,\Theta)}{\partial \Theta} P(\sigma|G,\Theta).$$

Note, that the gradient $\frac{\partial l(\Theta)}{\partial \Theta}$ is summed over all permutations σ. But, this can be calculated more efficiently in $O(E)$ by employing the Metropolis sampling from $P(\sigma|G,\Theta)$. The Kronecker graphs have a static and a temporal properties of real networks. Furthermore, fitting the parameters of the initiator matrix is very fast even for large networks.

Modeling local network structure

Given a snapshot of the network at the time t_1, we want to infer new interactions among the vertices of existing network at some future time t_2. By some score function $f(u,v)$ we map a score to the particular edge (u,v) in the network and propose a ranked list of all the missing edges by this score function $f(u,v)$ in decreasing order [75]. Various functions for the edge confidence have been proposed. Let us denote the set of neighbours of the vertex x by $\Gamma(x)$. The common neighbours function [76] returns the number of elements in the intersection of the set:

$$f(u,v) = \Gamma(u) \bigcap \Gamma(v).$$

The Adamic and Adar [77] method calculates the score as:

$$f(u,v) = \sum_{z \in \Gamma(u) \bigcap \Gamma(v)} \frac{1}{log|\Gamma(z)|}.$$

This measure weights rarer features more heavily by using log function. The preferential attachment [78] coefficient calculates the score as:

$$f(u,v) = |\Gamma(u)||\Gamma(v)|.$$

The Jacard's coefficient [79] calculates the score as:

$$f(u,v) = \frac{|\Gamma(u) \bigcap \Gamma(v)|}{|\Gamma(u) \bigcup \Gamma(v)|}.$$

The Katz measure [80] is a weighted sum over all the possible paths between the vertices (u,v):

$$f(u,v) = \sum_{l} \beta^{l} |paths_{l}(u,v)|.$$

In matrix terms the Katz measure between all pairs of the vertices is: $(I - \beta A)^{-1} - I$, where A is the adjacency matrix of the network. The commute time $C_{u,v}$ is a sum of the expected number of steps required that the random walker start at u and reaches v and comes back.

A Low-rank matrix approximation with the singular value decomposition $A_k = U_k S_k V_k^T$ of an adjacency matrix can also be used as one approach for an edge prediction. Many of these measures have outperformed the random predictor, just by using the topology properties [75].

Hierarchical Random Graphs (HRG) [81] are the general method for inferring a hierarchical network structure. The hierarchical structure is represented by a tree or a dendogram in which the lowest common ancestor represents the probability p_r of the edge between a pair of vertices in the network. The number of leaves in the dendogram is equal to the number of vertices in the network. We are interested in fitting the hierarchical model $(D, \{p_r\})$ with the real network G. This is accomplished by using a maximum likelihood method with the Monte Carlo sampling algorithm on the space of all possible dendograms $(D, \{p_r\})$. The likelihood between real network G and the HRG is:

$$L(D, \{p_r\}) = \prod_{r \in D} p_r^{E_r} (1 - p_r)^{L_r R_r - E_r},$$

where E_r is the number of edges in G whose vertices have r as the lowest common ancestor, L_r and R_r represent the number of leaves in the left and the right subtrees at the lowest common ancestor r. By this method we can sample dendograms proportional to their likelihood to generate a real network. The final result of this method is an ensemble of dendograms which are merged to the consensus dendogram. The hierarchical structure can be used for prediction of missing edges in the near future. We just output the ranked list of the edges that are missing in the original network G according to the corresponding edge probabilities.

Network structures at the level of groups

A community or a cluster or a subgroup is a subgraph whose vertices are connected more cohesively or densely than with the outside vertices. Different definitions of communities (clique, n-clique, k-plex, etc.) are possible. A clique is a subgraph where all vertices are connected with each other. The n-clique is a subgraph where all pairs of vertices have a geodesic distance less or equal to n. A k-plex is a maximal subgraph with m nodes where each vertex has $m-k$ neighbours in the subgraph.

The Kernighan-Lin algorithm [82] is a heuristic algorithm used for the graph bisection problem (division of vertices into two cohesive groups). This algorithm starts with an arbitrary division into two groups and searches over all pairs of vertices whose interchange would minimize the cut size.

Spectral graph partitioning uses the second eigenvector v_2 (Fiedler eigenvector) associated to the second lowest eigenvalue λ_2. The positive components in Fiedler's eigenvector represent vertices in the first subgraph, while other components represent vertices in the second subgraph [83].

A hierarchical clustering is used when the number of clusters is not known in advance. The aim is to divide vertices into clusters, such that vertices within the cluster are more closely related. This agglomerative hierarchical clustering starts by assigning each vertex its own cluster and iteratively merges the closest (similar) pairs of clusters into a single cluster. The hierarchical random graph model [81] is also one example of this technique.

The algorithm by Girvan and Newman [84] for community detection is based on iterative pruning of the edges with the highest betweenness, until the network breaks into components. Another very important community detection algorithms can be found in the review paper by Fortunato [85].

Appendix B

Fundamentals of epidemic modelling on networks

The field of classical mathematical epidemiology has a long history in modelling the epidemic processes [8] by using: the set of ordinary differential equations for macro level dynamics description of deterministic processes, the Markov chain theory for micro level dynamics description of stochastic processes and stochastic differential equations for macro level dynamics description of stochastic processes. But in this work, we are interested in modelling the stochastic epidemic processes on network structures, where we shortly describe four different theoretical approaches: (i) bond percolation approach, (i) individual-based mean field approach, (iii) degree-based mean field approach and (iv) message passing approach for stochastic SIR epidemic modelling on networks.

Bond percolation

The process of random removal of nodes or edges in a network is called the percolation process. The process of random removal of nodes or edges is called the site or bond percolation, respectively. The percolation theory studies the behaviour of connected components in random graphs. The removal probability is called the occupation probability and is denoted with ϕ.

Under the continuous SIR model, the probability that the disease propagates through an edge from infected node which stays infected τ amount of time is:

$$\phi = 1 - e^{-\beta\tau}. \tag{B.1}$$

Generally, the SIR process can exactly be mapped to the semi-directed percolation networks [17, 19]. But under the assumption that all nodes have approximately the same recovery time τ, the simple bond percolation process with the parameter ϕ to occupy the edge can approximate the final outcome of the SIR process [15, 16]. The occupied edge represents the edge along which the disease would be transmitted if it reaches the incident nodes. Now, if the disease starts at some random initial node, the final outcome of the epidemic is the set of nodes, which is connected to the initial node just by traversing the occupied edges.

Analytically, it is possible to calculate the average behaviour of the final outcome of the SIR process on the ensemble of graphs with predefined degree distribution p_k (configuration model). Let us now, define the average probability that a node is not connected to the giant component via its own specific edge with u. This happens either because the edge is not occupied (probability $1-\phi$) or the edge is occupied (probability ϕ) but then the incident node with excess degree k is not in giant component (probability u^k). If we average the result over all excess degrees in a network, we get:

$$u = \sum_{k=0}^{\infty} q_k(1-\phi+\phi u^k) = 1-\phi+\phi G_1(u), \tag{B.2}$$

where q_k denotes the probability of excess degree in a network and $G_1(u)$ its generating function. If by following an edge we come to the node with degree $k+1$, by definition its excess degree is k, as we do not account the traversed edge. The excess degree distribution is calculated from the regular degree distribution:

$$q_k = \frac{k+1}{\sum_i k_i} N p_{k+1} = \frac{(k+1)p_{k+1}}{\langle k \rangle} \tag{B.3}$$

or from generating functions:

$$G_1(x) = \frac{G_0(x)'}{G_0(1)'}, \tag{B.4}$$

where the $G_0(x) = \sum_k p_k x^k$ is the generating function of degree distribution p_k. By averaging over the degree distribution, we get the final size of epidemic $\langle X \rangle$:

$$\langle X \rangle = 1 - \sum_{k=0}^{\infty} p_k u^k = 1 - G_0(u). \tag{B.5}$$

The epidemic threshold is obtained when the curve $f(u) = u$ is tangent to $f(u) = 1-\phi+\phi G_1(u)$ at the point $u = 1$:

$$\left(\frac{\partial}{\partial u}(1-\phi+\phi G_1(u)) \right)_{u=1} = 1. \tag{B.6}$$

So the critical value is equal to:

$$\phi_c = \left(G_1(1)' \right)^{-1} = \left(\sum_k \frac{k(k+1)p_{k+1}}{\langle k \rangle} \right)^{-1} = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}. \tag{B.7}$$

Now, for random networks with Poisson degree distribution: $p_k = e^{-\lambda} \frac{\lambda^k}{k!}$, we get $\langle k \rangle = \lambda$,

$\langle k^2 \rangle = \lambda^2 + \lambda$ and the critical epidemic threshold is $\phi_c = \lambda^{-1}$. For networks with the power law degree distribution $p_k \propto k^{-\alpha}$, with exponents $2 < \alpha < 3$, we get a finite mean $\langle k \rangle$ with a second moment $\langle k^2 \rangle$ which diverges. Which implies that the critical threshold vanishes $\phi_c = 0$ (there is always an epidemic).

Individual-based Mean Field

The Individual-based Mean Field approach [86], assumes that the dynamical state of every node is independent of the state of neighbours. This implies that the expectation of node state product factorizes $\langle X_i X_j \rangle = \langle X_i \rangle \langle X_j \rangle$. This approach keeps the structure of the network by using the adjacency matrix A_{ij} which is static or quenched. The probability that a node i is in susceptible, infected or recovered state is denoted with: ρ_i^S, ρ_i^I and ρ_i^R, respectively. The random variables: S_i, I_i and R_i denote the Bernoulli random variables that the node i is some state. We can write the $2N$ equations for the probabilities of state variables:

$$\frac{d}{dt}\rho_i^S(t) = -\beta \sum_j A_{ij} \langle S_i I_j \rangle, \tag{B.8}$$

$$\frac{d}{dt}\rho_i^I(t) = \beta \sum_j A_{ij} \langle S_i I_j \rangle - \gamma \rho_i^I(t). \tag{B.9}$$

Due to the independence assumption and property of Bernoulli random variable, we get:

$$\frac{d}{dt}\rho_i^S(t) = -\beta \sum_j A_{ij} \rho_i^S(t) \rho_j^I(t), \tag{B.10}$$

$$\frac{d}{dt}\rho_i^I(t) = \beta \sum_j A_{ij} \rho_i^S(t) \rho_j^I(t) - \gamma \rho_i^I(t). \tag{B.11}$$

If we choose the initial conditions in a way that $\rho_i^S(0)$ tends to 1 (almost everyone is susceptible), $\rho_i^I(0)$ is small (small number of random initial infected nodes) and $\rho_i^R(0)$ is 0 (no one has recovered) then $\rho_i^S(0) = 1 - \rho_i^I(0) - \rho_i^R(0) = 1 - \rho_i^I(0)$. So, we can neglect the quadratic terms from the previous equation: $\beta \sum_j A_{ij} \rho_i^S(t) \rho_j^I(t) = \beta \sum_j A_{ij} (1 - \rho_i^I(0)) \rho_j^I(t) \approx \beta \sum_j A_{ij} \rho_j^I(t)$ and we get:

$$\frac{d}{dt}\rho_i^I(t) = \beta \sum_j A_{ij} \rho_j^I(t) - \gamma \rho_i^I(t) = \sum_j (\beta A_{ij} - \gamma \delta_{ij}) \rho_j^I(t), \tag{B.12}$$

where δ_{ij} is the Kronecker delta. Which can be written in matrix form as: $\frac{d}{dt}\rho^I(t) = \beta M \rho^I$, where M is the symmetric matrix: $A - \frac{\gamma}{\beta} I$. The solution of this system is written as a linear combination of eigenvectors $\vec{v}_i$ with associated eigenvalues Λ_i:

$$\vec{x}(t) = \sum_{i=1}^{n} a_i(0)\vec{v}_i e^{(\beta\Lambda_i - \gamma)t}. \tag{B.13}$$

The dominant term is with the leading positive eigenvalue Λ_1 and the epidemic threshold [87] is equal to:

$$\phi_c = \frac{\beta}{\gamma} = \frac{1}{\Lambda_1}. \tag{B.14}$$

Note, that the more dense networks have bigger the leading eigenvalue Λ_1 and therefore the smaller epidemic threshold.

But, note that this approximation neglects the correlation between the neighbouring states and this is the reason for the introduction of pair-approximation approaches by modelling the $\langle X_i X_j X_k \rangle$ as relevant quantities [88]. Here, we only sketch the idea for a pair-approximation approach.

$$\frac{d}{dt}\langle S_i I_j \rangle = \beta \sum_{k \neq i} A_{jk} \langle S_i S_j I_k \rangle - \beta \sum_{l \neq j} A_{il} \langle I_l S_i I_j \rangle - \beta \langle S_i I_j \rangle, \tag{B.15}$$

where the first term denotes the contributions when neighbour k infects susceptible node j, the second term denotes contributions when infected neighbour l infects node i and the last term the contribution when infected node j infects node i. Now, with the use of Bayes theorem we get:

$$\langle S_i S_j I_k \rangle = P(i,j \in S, k \in I) = P(i,j \in S) P(k \in I | i,j \in S), \tag{B.16}$$

where we use the assumption that the state of node k is independent of node i: $P(k \in I | i,j \in S) = P(k \in I | j \in S)$ and we get:

$$\langle S_i S_j I_k \rangle = \langle S_i S_j \rangle \frac{\langle S_j I_k \rangle}{\langle S_j \rangle}. \tag{B.17}$$

Similarly, the three point expectation $\langle I_l S_i I_j \rangle$:

$$\langle I_l S_i I_j \rangle = \langle S_i I_j \rangle \frac{\langle I_l S_i \rangle}{\langle S_i \rangle}. \tag{B.18}$$

This method of approximating three point moment with the combination of two and one point moments is called the moment closure method.

Degree-based Mean Field

Degree-based mean field approach [89] [90] assumes that the nodes with the same degree k are statistically equivalent to the spreading process. This approach seriously reduces the number of degrees of freedom in a system. This approach assumes replacement of adjacency matrix A_{ij} with the ensemble average $\overline{A}_{ij}$ (annealed network approximation). This approach has three dynamical variables: $\rho_k^S(t)$, $\rho_k^I(t)$ and $\rho_k^R(t)$, which represent the probability that a node with degree k is susceptible, infected or recovered in time t. Let us now, derive the equation for the probability that the susceptible node with degree k gets infected between t and $t+dt$ time. First, it has to get the disease from its neighbours and the average probability that the neighbour is infected is:

$$\Gamma_k(t) = \sum_{k'} P(k'|k)\rho_{k'}^I(t). \tag{B.19}$$

The probability that the disease spreads from a single neighbour during dt time is: $\beta\Gamma_k(t)dt$ and the expected probability that it gets the disease from either of k neighbours independently is $k\beta\Gamma_k(t)dt$. And in order write the rate of change for $\rho_k^S(t)$, we require that the node itself is susceptible, which happens with the probability $\rho_k^S(t)$:

$$\frac{d}{dt}\rho_k^S(t) = -k\beta\Gamma_k(t)\rho_k^S(t), \tag{B.20}$$

similarly we write rate of change for other quantities:

$$\frac{d}{dt}\rho_k^I(t) = k\beta\Gamma_k(t)\rho_k^S(t) - \gamma\rho_k^I(t), \tag{B.21}$$

$$\frac{d}{dt}\rho_k^R(t) = \gamma\rho_k^I(t). \tag{B.22}$$

In a case of uncorrelated networks the average probability that the neighbour is infected is:

$$\Gamma_k(t) = \sum_k q_k\rho_k^I(t) = \sum_k q_k(1-\rho_k^R(t)-\rho_k^S(t)) = 1 - \underbrace{\sum_k q_k\rho_k^R(t)}_{w(t)} - \sum_k q_k\rho_k^S(t), \tag{B.23}$$

where q_k is the excess degree and $w(t)$ is the average probability that a neighbour is recovered.

$$\frac{d}{dt}w(t) = \sum_k q_k\frac{d}{dt}\rho_k^R(t) = \gamma\sum_k q_k\rho_k^I(t) = \gamma\Gamma(t), \tag{B.24}$$

From this we eliminate Γ from $\frac{d}{dt}\rho_k^S(t)$:

$$\frac{d}{dt}\rho_k^S(t) = -k\frac{\beta}{\gamma}\left(\frac{d}{dt}w(t)\right)\rho_k^S(t), \tag{B.25}$$

and by integrating we get:

$$\rho_k^S(t) = \rho^S(0)e^{-k\frac{\beta}{\gamma}w(t)} = \rho^S(0)(u(t))^k, \tag{B.26}$$

where $u(t) = e^{-\frac{\beta}{\gamma}w(t)}$. Then, we turn the equation (B.23) as a function of $u(t)$:

$$\Gamma_k(t) = 1 + \frac{\gamma}{\beta}ln(u(t)) - \sum_k q_k u(t)^k = 1 + \frac{\gamma}{\beta}ln(u(t)) - G_1(u(t)). \tag{B.27}$$

The rate of change of $u(t)$ is a first order differential equation:

$$\frac{d}{dt}u(t) = -\beta u(t)\left(1 + \frac{\gamma}{\beta}ln(u(t)) - G_1(u(t))\right). \tag{B.28}$$

When we have $u(t)$ as a solution of the previous equation the probability of the node being susceptible is:

$$\rho^S(t) = \sum_k p_k\rho_k^S(t) = \rho^S(0)\sum_k p_k(u(t))^k = \rho^S(0)G_0(u(t)). \tag{B.29}$$

Other quantities $\rho^I(t)$, $\rho^R(t)$ can also be derived. The outbreak size is given in the time limit:

$$\rho^R(\infty) = 1 - \rho^S(\infty) = 1 - \rho^S(0)G_0(u(\infty)). \tag{B.30}$$

This model gives similar epidemic threshold $\phi_c = \frac{1}{G_1'(1)}$.

Message passing – belief propagation

Belief propagation, message-passing and cavity methods [91] belong to the same class of inference methods used in different fields like information theory, statistical physics and artificial intelligence to calculate marginal distributions of random variables. Here, we will present the basic idea behind the message passing algorithms for continuous SIR epidemics on tree networks [24] because the discrete SIR version of message passing equations are very similar [25]. We assume that the transmission probability of disease from infected node to susceptible node in $d\tau$ time is $s(\tau)d\tau$. The probability of recovery of infected node in $d\tau$ time is $r(\tau)d\tau$. So, the total probability of transmission between time τ and $\tau + d\tau$ is the intersection of event that the node recovered after τ time and that the transmission occurred between τ and $\tau + d\tau$ time:

$$f(\tau)d\tau = s(\tau)d\tau \int_{\tau}^{\infty} r(\tau')d\tau'. \tag{B.31}$$

Now, we are interested in the probability $H^{i\leftarrow j}(t)$ that the node j does not propagate the disease to node i up to the time t. This can happen either because the node j transmits the infection message after the interval t or the node j is about to transmit the infection message in the time $\tau \leq t$ but gets the infection from his neighbours in time $t' > t - \tau$, which is too late to be able to transmit it before time t. This is quantified as the sum of two disjoint probabilities:

$$H^{i\leftarrow j}(t) = \left(1 - \int_0^t f(\tau)d\tau\right) + P(S_j(0)) \left(\int_0^t f(\tau) \prod_{k\in N(j)\setminus i} H^{j\leftarrow k}(t-\tau)d\tau\right), \tag{B.32}$$

where $P(S_j(0))$ is the initial probability that a node j is susceptible at the start of epidemics. The quantity $H^{i\leftarrow j}(t)$ is the message that is being transmitted between nodes in a network. From this quantity we can calculate the relevant marginal epidemiological probabilities: $P(S_i(t))$ that a node is in state S at time t.

$$P(S_i(t)) = P(S_i(0)) \prod_{j\in N(i)} H^{i\leftarrow j}(t). \tag{B.33}$$

Other quantities $P(I_i(t))$ or $P(R_i(t))$ can also be derived.

Appendix C

Alternative SIR algorithms

The implementations of the epidemic algorithms used in this thesis are publicly available at the GitHub [*]. In this chapter, we give two alternative SIR algorithms: (i) the event-based SIR algorithm and (ii) the lazy-recovery SIR algorithm. First, we start with the event-based SIR algorithm, which is hybrid between the Naive SIR ant the FastSIR algorithm. This algorithm follows evolution of the SIR epidemics by following transmission events in time ascending order.

Now, we show the lazy-recovery SIR algorithm, which is suitable for temporal networks, where edges appear and disappear in time. The previous algorithms like: the event-based SIR algorithm and the FastSIR algorithm are optimized for static networks and therefore can not be applied on a temporal case and the NaiveSIR algorithm in principle can be modified for a temporal network. Note, that the lazy-recovery SIR algorithm was used in a case study of detecting the source of STI disease on empirical temporal network. The lazy-recovery SIR algorithm is linear in number of contacts, since we employ a lazy recovery technique where we test whether node has recovered only when it has a contact with other nodes. In order to enable this lazy recovery we have to recalculate the recovery probability by amount of time the node was not probed for recovery.

Markov Chain Monte Carlo algorithm

In this section, we explain how to construct Markov Chain Monte Carlo (MCMC) algorithms for epidemic simulation of a discrete SIR process on networks. Usually, if we need to get the statistical properties of Monte Carlo SIR process, we run *n* independent simulations from same initial conditions and then calculate statistical properties over *n* realizations or samples. The basic idea of MCMC method is to construct new samples or realizations from previous ones with some stochastic method. We will now describe the MCMC method for sampling realization

---

[*] https://github.com/ninoaf/epidemics

Algorithm 9 The event-based SIR algorithm

```
Input: $(G, C_p, C_q, I, S, \theta, T)$ where $G$ is contact network, $C_p$ is a cumulative distribution for disease transmission time, $C_q$ is a cumulative distribution for node recovery time, $I$ is a array of transmissions events - $I(t)$ is a list of nodes which have been scheduled for becoming infective at time $t$, $S(v)$ is an array indicator of susceptible nodes, $\theta$ is the initially infected node and $T$ is a stopping time for a simulation.
Output: array indicator of susceptible nodes $S(v)$ prior to $T$
push($I(0), \theta$)
for time $t = 0$ to T do
  event-list = $I(t)$
  for each node $u$ in event-list do
    if $S(u)$ is equal to 1 then
      $S(u) = 0$ // infect node u
      $r_u \sim C_q$ // sample recovery time from CDF
      for each neighbouring node $v$ of $u$ in $G$ do
        $t_{uv} \sim C_p$ // sample transmission time from CDF
        if $t_{uv} \leq r_u$ then
          push($I(t + t_{uv}), v$)
        end if
      end for
    end if
  end for
end for
output $S(v)$
```

Algorithm 10 The lazy-recovery SIR algorithm

```
**Input:** $(G,S,p,q,\theta,T)$ where $G$ is contact network: array of triplets: $G(i) = (u_i,v_i,t_i)$ node $u_i$ is connected to node $v_i$ at time $t_i$, $S(v)$ is an array indicator of susceptible nodes, $\theta$ is the initially infected node, $p$ transmission probability in one discrete step, $q$ recovery in one discrete time and $T$ is a stopping time for a simulation.
**Output:** array indicator of susceptible nodes $S(v)$ prior to $T$
sort contacts in $G$ ascending in time
$S(\theta) = 0$
$I_t(v) = 0$, $I_t(\theta) = 1$ Initialize array indicator of infective nodes in time
$\psi(v) = 0$ Initialize array where we store probe recovery time for each node
while $i <$ size(G(i)) or $t_i \leq T$ do
  if $(I_t(u_i) == 1)$ contact $u_i$ is infective then
    $\delta t = t_i - \psi(u_i)$ amount of time since last recovery probe time
    if $(rand() \leq 1-(1-q)^{\delta t})$ then
      $I_t(u_i) = 0$ node has recovered since last probing time
    end if
    $\psi(u_i) = t_i$ update probe recovery time
  end if
  if $(I_t(v_i) == 1)$ contact $v_i$ is infective then
    $\delta t = t_i - \psi(v_i)$ amount of time since last recovery probe time
    if $(rand() \leq 1-(1-q)^{\delta t})$ then
      $I_t(v_i) = 0$ node has recovered since last probing time
    end if
    $\psi(v_i) = t_i$ update probe recovery time
  end if
  if $(I_t(u_i) == 1)$ and $(S(v_i) == 1)$ and $(rand() \leq p)$ then
    $S(v_i) = 0$, $I_t(v_i) = 1$ node $v_i$ becomes infected from node $u_i$
    $\psi(v_i) = t_i + 1$ update probe recovery time
  end if
  if $(I_t(v_i) == 1)$ and $(S(u_i) == 1)$ and $(rand() \leq p)$ then
    $S(u_i) = 0$, $I_t(u_i) = 1$ node $u_i$ becomes infected from node $v_i$
    $\psi(u_i) = t_i + 1$ update probe recovery time
  end if
  ++i next contact $(u_i,v_i,t_i)$
end while
output $S(v)$
```

of the discrete SIR epidemic process with parameters λ up to time threshold T. Let us define the random vector $\vec{R}(t) = (R(1), R(2), ..., R(N))$ that indicates which nodes got infected up to a time moment t. The random variable $R(i)$ is a Bernoulli random variable, which has the value of 1 if the node i got infected before time t from the start of the epidemic process SIR with parameters λ and the value of 0 otherwise. The ideas is the following: (i) first, we normally run Monte Carlo simulation up to time T to obtain realization: $\vec{r}_1(T)$ of random vector $\vec{R}_\lambda$, (ii) we memorize the history (trajectory) of the current realization in time: $\{\vec{r}_1(t) : t = 0..T\}$, (ii) then we sample the return time moment $t^* < T$ and (iv) construct new realization sample $\vec{r}_2(T)$ as a result of Monte Carlo simulation for $T - t^*$ time with initial conditions from realization $\vec{r}_1(t^*)$. We have to make sure, that this procedure does not affect the probability over realization space $P(\vec{R}(t))$.

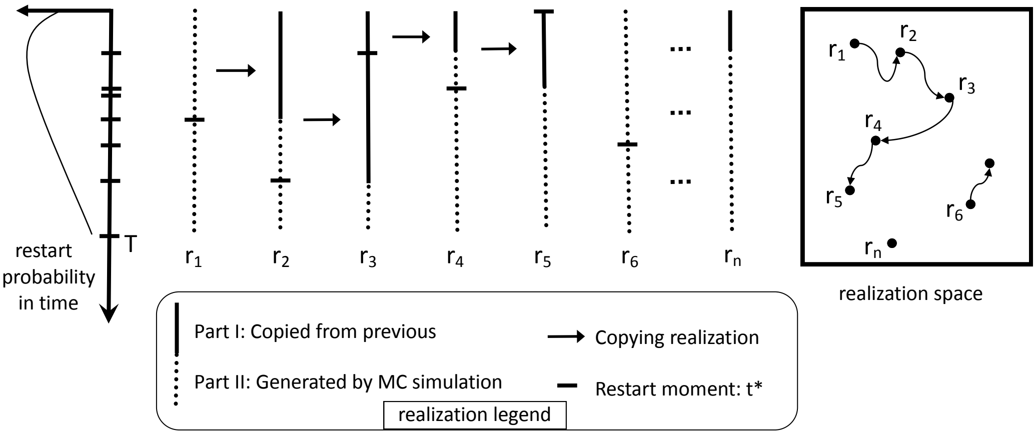


Figure C.1: Diagram of Markov Chain Monte Carlo method for sampling realizations of the SIR epidemic process on network. The first realization $\vec{r}_1(T)$ is sampled with normal Monte Carlo SIR simulation, denoted with dashed line. Next realization $\vec{r}_2(T)$ is obtained by sampling a restart time t^*, setting the previous realization at time t^* as new initial condition $\vec{r}_1(t^*)$ and continuing Monte Carlo simulation for $T - t^*$ time to get $\vec{r}_2(T)$ and the process continues. Note, that when the sampled restart time $t^* = 0$, there is no correlation with the previous sample, e.g. $\vec{r}_6(T)$ is independent of the previous samples. This process can be visualized in realization space, where each realization is one point and Markov transitions between points are denoted with arrows.

Bibliography

[1] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review* **45** no. 2, (2003) 167–256.

[2] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Critical phenomena in complex networks," *Rev. Mod. Phys.* **80** no. 4, (2008) 1275–1335.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks : Structure and dynamics," *Phys. Rep.* **424** no. 4-5, (2006) 175–308.

[4] A. Vespignani, "Modelling dynamical processes in complex socio-technical systems," *Nat. Phys.* **8** no. 1, (2012) 32–39.

[5] W. O. Kermack and A. G. McKendrick, "Contributions to the mathematical theory of epidemics," *Proc. Roy. Soc. London Ser. A* **115** (1927) 700–721.

[6] G. Serazzi and S. Zanero, "Computer virus propagation models," in *Performance Tools and Applications to Networked Systems*, M. Calzarossa and E. Gelenbe, eds., vol. 2965 of *Lecture Notes in Computer Science*, pp. 26–50. Springer Berlin Heidelberg, 2004.

[7] Y. Moreno, M. Nekovee, and A. F. Pacheco, "Dynamics of rumor spreading in complex networks," *Phys. Rev. E* **69** (2004) 066130.

[8] H. Andersson and T. Britton, *Stochastic Epidemic Models and Their Statistical Analysis*. Lecture Notes in Statistics. Springer, 2000.

[9] R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani, "Epidemic processes in complex networks," `arXiv:1408.2701 [physics.soc-ph]`.

[10] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review* **42** (2000) 599–653.

[11] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature* **393** no. 6684, (1998) 440–442.

[12] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science* **286** no. 5439, (1999) 509–512.

[13] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, “Structure of growing networks with preferential linking,” *Phys. Rev. Lett.* **85** no. 21, (2000) 4633–4636.

[14] R. Pastor-Satorras and A. Vespignani, “Epidemic spreading in scale-free networks,” *Phys. Rev. Lett.* **86** no. 14, (2001) 3200–3203.

[15] D. Mollison, “Spatial contact models for ecological and epidemic spread,” *J R Stat Soc Series B* **39** no. 3, (1977) 283–326.

[16] P. Grassberger, “On the critical behavior of the general epidemic process and dynamical percolation,” *Mathematical Biosciences* **63** no. 2, (1983) 157 – 172.

[17] L. Meyers, M. Newman, and B. Pourbohloul, “Predicting epidemics on directed contact networks,” *Journal of Theoretical Biology* **240** no. 3, (2006) 400–418.

[18] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Network robustness and fragility: Percolation on random graphs,” *Phys. Rev. Lett.* **85** (2000) 5468–5471.

[19] E. Kenah and J. M. Robins, “Second look at the spread of epidemics on networks,” *Phys. Rev. E* **76** (2007) 036113.

[20] M. E. J. Newman and R. M. Ziff, “Fast monte carlo algorithm for site or bond percolation,” *Phys. Rev. E* **64** (2001) 016706.

[21] C. Castellano and R. Pastor-Satorras, “Thresholds for epidemic spreading in networks.,” *Phys. Rev. Lett.* **105** no. 21, (2010) 218701.

[22] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, “Epidemic spreading in real networks: an eigenvalue viewpoint,” in *Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on*, pp. 25–34. 2003.

[23] A. Lančić, N. Antulov-Fantulin, M. Šikić, and H. Štefančić, “Phase diagram of epidemic spreading — unimodal vs. bimodal probability distributions,” *Physica A: Statistical Mechanics and its Applications* **390** no. 1, (2011) 65 – 76.

[24] B. Karrer and M. E. J. Newman, “Message passing approach for general epidemic models,” *Phys. Rev. E* **82** no. 1, (2010) 016101.

[25] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, “Inferring the origin of an epidemic with a dynamic message-passing algorithm,” *Phys. Rev. E* **90** (2014) 012801.

[26] J. P. Gleeson, “Binary-state dynamics on complex networks: Pair approximation and beyond,” *Phys. Rev. X* **3** (2013) 021004.

[27] D. T. Gillespie, “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions,” *Journal of Computational Physics* **22** no. 4, (1976) 403 – 434.

[28] V. Colizza, A. Barrat, M. Barthelemy, A.-J. Valleron, and A. Vespignani, “Modeling the Worldwide Spread of Pandemic Influenza: Baseline Case and Containment Interventions,” *PLoS Med* **4** no. 1, (2007) e13.

[29] W. Broeck, C. Gioannini, B. Goncalves, M. Quaggiotto, V. Colizza, and A. Vespignani, “The gleamviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale,” *BMC Infectious Diseases 11, 37 (2011)* (2011) .

[30] K. R. Bisset, J. Chen, X. Feng, V. A. Kumar, and M. V. Marathe, “Epifast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems,” in *Proceedings of the 23rd international conference on Supercomputing*, ICS ’09, pp. 430–439. ACM, New York, NY, USA, 2009.

[31] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe, “EpiSimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks,” in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC ’08. IEEE Press, Piscataway, NJ, USA, 2008.

[32] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, “Modelling disease outbreaks in realistic urban social networks,” *Nature* **429** no. 6988, (2004) 180–184.

[33] D. Shah and T. Zaman, “Detecting sources of computer viruses in networks: theory and experiment,” in *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS ’10, pp. 203–214. ACM, New York, NY, USA, 2010.

[34] W. Dong, W. Zhang, and C. W. Tan, “Rooting out the rumor culprit from suspects,” in *Proceedings of the IEEE Intl. Symp. on Information Theory 2013*. 2013.

[35] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, “Rumor source detection with multiple observations: Fundamental limits and algorithms,” *SIGMETRICS Perform. Eval. Rev.* **42** no. 1, (June, 2014) 1–13.

[36] K. Zhu and L. Ying, “Information source detection in the SIR model: A sample path based approach,” in *Proceedings of the Information Theory and Applications 2013: San Diego, CA, USA*, pp. 1–9. 2013.

[37] P. C. Pinto, P. Thiran, and M. Vetterli, “Locating the Source of Diffusion in Large-Scale Networks,” *Phys. Rev. Lett.* **109** (2012) 068702+.

[38] F. Altarelli, A. Braunstein, L. Dall’Asta, A. Lage-Castellanos, and R. Zecchina, “Bayesian inference of epidemics on networks via belief propagation,” *Phys. Rev. Lett.* **112** no. 11, (2014) 118701.

[39] N. Antulov-Fantulin, A. Lancic, H. Stefancic, M. Sikic, and T. Smuc, “Statistical inference framework for source detection of contagion processes on arbitrary network structures,” in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference*, pp. 78–83. 2014.

[40] C. H. Comin and L. da Fontoura Costa, “Identifying the starting point of a spreading process in complex networks,” *Phys. Rev. E* **84** (2011) 056105.

[41] D. Brockmann and D. Helbing, “The Hidden Geometry of Complex, Network-Driven Contagion Phenomena,” *Science* **342** no. 6164, (2013) 1337–1342.

[42] B. A. Prakash, J. Vreeken, and C. Faloutsos, “Spotting culprits in epidemics: How many and which ones?,” in *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM ’12, pp. 11–20. IEEE Computer Society, Washington, DC, USA, 2012.

[43] W. Zang, P. Zhang, C. Zhou, and L. Guo, “Discovering multiple diffusion source nodes in social networks,” *Procedia Computer Science* **29** (2014) 443 – 452. 2014 International Conference on Computational Science.

[44] Z. Chen, K. Zhu, and L. Ying, “Detecting multiple information sources in networks under the sir model,” in *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pp. 1–4. 2014.

[45] W. Krauth, *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Physics. Oxford University Press, UK, 2006.

[46] N. Antulov-Fantulin, A. Lancic, H. Stefancic, and M. Sikic, “FastSIR algorithm: A fast algorithm for the simulation of the epidemic spread in large networks by using the susceptible–infected–recovered compartment model,” *Information Sciences* **239** (2013) 226 – 240.

[47] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001.

[48] M. Šikić, A. Lančić, N. Antulov-Fantulin, and H. Štefančić, "Epidemic centrality — is there an underestimated epidemic impact of network peripheral nodes?," *The Europ. Phys. Journal B* **86** no. 10, (2013) .

[49] J. S. Vitter, "Faster methods for random sampling," *Commun. ACM* **27** no. 7, (1984) 703–718.

[50] M. Sosic and M. Sikic, "Cuda implementation of the algorithm for simulating the epidemic spreading over large networks," in *MIPRO, 2012 Proceedings of the 35th International Convention*, pp. 1807–1810. 2012.

[51] G. Csardi and T. Nepusz, "The igraph Software Package for Complex Network Research," *InterJournal* **Complex Systems** (2006) .

[52] T. Granlund and the GMP development team, *GNU MP: The GNU Multiple Precision Arithmetic Library*, 2012. `http://gmplib.org/`.

[53] M. E. Newman, "The structure of scientific collaboration networks," *Proc. Natl. Acad. Sci. U S A* **98** no. 2, (2001) 404–409.

[54] M. E. J. Newman, "Internet at the level of autonomous systems – network data set 2006, unpublished, http://www-personal.umich.edu/ mejn/netdata/, 2015.".

[55] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *In Proceedings of the 5th ACM/USENIX Internet Measurement Conference IMC'07*. 2007.

[56] M. Newman, *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[57] P. Wegner, "A technique for counting ones in a binary computer," *Commun. ACM* **3** no. 5, (1960) 322–.

[58] N. Antulov-Fantulin, A. Lancic, H. Stefancic, T. Smuc, and M. Sikic, "Detectability limits of epidemic sources in networks," `arXiv:1406.2909 [cs.SI]`.

[59] L. E. C. Rocha, F. Liljeros, and P. Holme, "Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts," *PLoS Comput. Biol* **7** no. 3, (2011) e1001109.

[60] O. Woolley-Meza, C. Thiemann, D. Grady, J. J. Lee, H. Seebens, B. Blasius, and D. Brockmann, "Complexity in human transportation networks: a comparative analysis of worldwide air transportation and global cargo-ship movements," *The Eur. Phys. Journal B - Condensed Matter and Complex Systems* **84** no. 4, (2011) 589–600.

[61] M. R. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and Related Properties of Random Sequences and Processes*. Springer Series in Statistics. Springer Verlag, 1983.

[62] B. Bollobás, *Modern graph theory*. Graduate texts in mathematics. Springer, New York, Berlin, Paris, 1998.

[63] P. Bonacich, "Power and centrality: A family of measures," *American Journal of Sociology* **92** no. 5, (1987) 1170–1182.

[64] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM* **46** no. 5, (1999) 604–632.

[65] M. E. J. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.* **89** no. 20, (2002) 208701.

[66] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, New York, NY, USA, 2004.

[67] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Comput. Netw.* **33** (2000) 309–320.

[68] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature* **411** no. 6833, (2001) 41–42, `arXiv:cond-mat/0105306`.

[69] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi, "The large-scale organization of metabolic networks," *Nature* **407** no. 6804, (2000) 651–654.

[70] R. De Castro and J. Grossman, "Famous trails to Paul Erdős," *The Mathematical Intelligencer* **21** no. 3, (1999) 51–53.

[71] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, "Classes of small-world networks," *Proc. Natl. Acad. Sci. U S A* **97** no. 21, (2000) 11149–11152.

[72] P. Erdős and A. Rényi, "On the evolution of random graphs," in *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pp. 17–61. 1960.

[73] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proceedings of the Eleventh ACM*

SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD ’05, pp. 177–187. ACM, New York, NY, USA, 2005.

[74] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, “Kronecker graphs: An approach to modeling networks,” *J. Mach. Learn. Res.* **11** (2010) 985–1042.

[75] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *J. Am. Soc. Inf. Sci. Technol.* **58** no. 7, (2007) 1019–1031.

[76] M. Newman, “Clustering and preferential attachment in growing networks,” *Phys. Rev. E* **64** no. 2, (2001) 025102.

[77] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social Networks* **25** (2001) 211–230.

[78] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, “Evolution of the social network of scientific collaborations,” *Physica A: Statistical Mechanics and its Applications* **311** no. 3-4, (2002) 590 – 614.

[79] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[80] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika* **18** no. 1, (1953) 39–43.

[81] A. Clauset, C. Moore, and M. E. J. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature* **453** (2008) 98–101.

[82] B. Kernighan and S. Lin, “An Efficient Heuristic Procedure for Partitioning Graphs,” *The Bell Systems Technical Journal* **49** no. 2, (1970) .

[83] M. Fiedler, “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory,” *Czechoslovak Mathematical Journal* **25** no. 4, (1975) 619–633.

[84] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E* **69** no. 2, (2004) 026113.

[85] S. Fortunato, “Community detection in graphs,” *Phys. Rept.* **486** no. 3-5, (2010) 75 – 174.

[86] K. Sharkey, “Deterministic epidemiological models at the individual level,” *Journal of Mathematical Biology* **57** no. 3, (2008) 311–331.

[87] B. Prakash, D. Chakrabarti, N. Valler, M. Faloutsos, and C. Faloutsos, “Threshold conditions for arbitrary cascade models on arbitrary networks,” *Knowledge and Information Systems* **33** no. 3, (2012) 549–575.

[88] K. Sharkey, I. Kiss, R. Wilkinson, and P. Simon, “Exact equations for SIR epidemics on tree graphs,” *Bulletin of Mathematical Biology* (2013) 1–32.

[89] M. Boguñá, C. Castellano, and R. Pastor-Satorras, “Langevin approach for the dynamics of the contact process on annealed scale-free networks,” *Phys. Rev. E* **79** no. 3, (2009) .

[90] Y. Moreno, R. Pastor-Satorras, and A. Vespignani, “Epidemic outbreaks in complex heterogeneous networks,” *The Eur. Phys. J. B - Condensed Matter and Complex Systems* **26** no. 4, (2002) 521–529.

[91] M. Mezard and A. Montanari, *Information, Physics, and Computation.* Oxford University Press, Inc., New York, NY, USA, 2009.

Biography

Nino Antulov-Fantulin was born on 22^{nd} of November 1986 at Slavonski Brod in Croatia. He finished undergraduate study in Computing, profile Computer Science in 2008 with Bachelor thesis title "Influence of epidemic spread on network structure" and the graduate study in Computing, profile Computer Science in 2010 with the Master thesis title "Protein Docking tool: rotation and scoring modules" at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He got employed at the Rudjer Boskovic Institute in Zagreb in 2010 as a research assistant working in data mining, machine learning and complex networks. In the same year, he enrolled in the post-graduate study of Computer Science at the Faculty of Electrical Engineering and Computing, where he was also a teaching assistant for the next 3 years. At the institute, he participated in different EU projects: "An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science", "Forecasting Financial Crisis", "Foundational Research on Multilevel Complex Networks and Systems" and was a research intern at the Robert-Koch Institute for epidemiology in Berlin in October and November 2013.

List of Publications

The research work from this thesis is from the publications denoted with the bold font.

Journal papers

1. **Antulov-Fantulin, N., Lancic, A., Stefancic, H. and Sikic, M., FastSIR Algorithm: "A Fast Algorithm for simulation of epidemic spread in large networks by using SIR compartment model", Information Sciences 239 (2013) , 226-240.**
2. **Antulov-Fantulin, N., Lancic, A., Smuc, T., Stefancic, H. and Sikic, M., "Identification of Patient Zero in Static and Temporal Networks - Robustness and Limitations", Phy. Rev. Lett. 114, 248701 (2015).**
3. Lančić, A., Antulov-Fantulin, N., Šikić, M. and Štefančić, H., "Phase diagram of epidemic spreading – unimodal vs. bimodal probability distributions", Physica A: Statistical Mechanics and its Applications 390 (2011) 65-76.

4. Šikić, M., Lančić, A., Antulov-Fantulin, N. and Štefančić, H., "Epidemic centrality and the underestimated epidemic impact on network peripheral nodes", The European Physical Journal B (2013), 86:440.
5. Piškorec, M., Antulov-Fantulin, N., Kralj Novak, P., Mozetič, I., Grčar, M., Vodenska, I., Šmuc, T., "Cohesiveness in Financial News and its Relation to Market Volatility", Scientific Reports (2014), 4:5038.

Conference proceedings

1. **Antulov-Fantulin, N., Lancic, A. Stefancic, H., Sikic, M., Smuc, M., Statistical inference framework for source detection of contagion processes on arbitrary network structures, Proceedings of 2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, London, pp: 78-83.**
2. Antulov-Fantulin, N., Bošnjak, M., Zlatic, V., Grcar, M., Smuc, T., "Synthetic sequence generator for recommender systems - memory biased random walk on sequence multilayer network", In Proceedings of 17th International Conference Discovery Science 2014, Bled, Lecture Notes in Computer Science, Vol. 8777, pp 25-36.
3. Mihelčić, M., Antulov-Fantulin, N., Bošnjak, M., Šmuc, T., "Extending RapidMiner with recommender systems algorithms", RapidMiner Community Meeting and Conference 2012, Budapest, pp: 63-74.
4. Bošnjak, M., Antulov-Fantulin, N. Šmuc, T. and Gamberger, D., "Constructing recommender systems workflow templates in RapidMiner", Proceedings of the RapidMiner Community Meeting And Conference 2011, Dublin, pp: 101-112.
5. Antulov-Fantulin, N., Bošnjak, M., Žnidaršič, M., Grčar, M., Morzy, M., Šmuc, T., "ECML/PKDD 2011 Discovery Challenge Overview", In Proc.of ECML-PKDD 2011 Discovery Challenge Workshop, Athens, pp 7-20.
6. Piškorec, M., Antulov-Fantulin, N., Ćurić, J., Dragoljević, O., Ivanac, V. and Karlović, L., "Computer vision system for the chess game reconstruction", Proceedings of the 34rd International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, pp: 263-269.
7. Sović, I., Antulov-Fantulin, N. Čanadi, I., Piškorec, M. and Šikić, M., "Parallel Protein Docking Tool", MIPRO 2010, Proceedings of the 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, pp: 1333-1338.

Publications under review

1. **Antulov-Fantulin, N., Lancic, A., Sikic, M., Smuc, T. and Stefancic, H. , "Multiple source epidemic detection by statistical inference on complex networks".**
2. Piskorec, M., Antulov-Fantulin, N., Miholic, I., Sikic, M., "Modeling of peer and external influence in online social network during December 1st 2013 referendum in Croatia".

Talks

1. **Antulov-Fantulin, N., Lancic, A., Smuc, T., Stefancic, H. and Sikic, M., "Detectability of epidemic sources in static and temporal networks", talk at the European Conference on Complex Systems 2014, Lucca, Italy.**
2. **Antulov-Fantulin, N., Lancic, A., Smuc, T., Stefancic, H. and Sikic, M., "Statistical inference of epidemic sources in bipartite temporal networks", talk at SINM NetSci 2014 satellite, in June 2014, Berkeley, USA.**
3. **Antulov-Fantulin, N., Lancic, A., Stefancic, H., Sikic, M. and Smuc, T., "Limits of Epidemic Source Localization in Complex Networks", talk at NetSci 2013, International School and Conference on Network Science 2013, Copenhagen, Denmark.**

Životopis

Nino Antulov-Fantulin rođen je 22. studenog 1986. u Slavonskom Brodu, Hrvatska. Završio je preddiplomski studij računarstva, profil računarska znanost 2008. godine sa završnim radom na temu: "Utjecaj zaraze na svojstva kompleksne mreže" i diplomski studij računarstva, profil računarska znanost 2010. godine sa diplomskim radom na temu "Alat za prianjanje proteina: moduli za rotaciju i vrednovanje" na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Zaposlio se 2010. godine na Institutu Ruđer Bošković u Zagrebu kao znanstveni novak, radeći na problemima dubinske analize podataka, strojnog učenja i kompleksnih mreža. Iste godine upisuje poslijediplomski studij računarstva na Fakultetu elektrotehnike i računarstva, gdje naredne 3 godine radi kao asistent u nastavi. Tijekom rada na institutu, sudjeluje na raznim europskim projektima: "An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science", "Forecasting Financial Crisis", "Foundational Research on Multilevel Complex Networks and Systems" i sudjeluje u znanstvenom posjetu i radu na epidemiološkom Institutu Robert-Koch u Berlinu u mjesecu listopadu i studenom 2013. godine.