IMT School for Advanced Studies, Lucca
Lucca, Italy

The Bitcoin Transaction Networks

PhD in Institutions, Markets and Technologies
Curriculum in Economics, Management and Data Science

XXXII Cycle

By

Nicolò Vallarano

2021

The dissertation of Nicolò Vallarano is approved.

PhD Program Coordinator: Prof. Massimo Riccaboni, IMT School for Advanced Studies Lucca

Advisor: Prof. Guido Caldarelli, University of Venice 'Ca' Foscari'

Co-Advisor: Dr. Tiziano Squartini, IMT School for Advanced Studies Lucca

The dissertation of Nicolò Vallarano has been reviewed by:

Prof. Fabio Caccioli, University College London

Prof. Nicola Dimitri, University of Siena

Dr. Angelo Facchini, IMT School for Advanced Studies Lucca

Prof. Silvia Giordano, SUPSI

IMT School for Advanced Studies Lucca
2021

Contents

List of Figures

List of Tables

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

First, I would like to thank my supervisor, Tiziano Squartini, whose expertise was invaluable in formulating the research questions and devise the most suitable techniques to address them: your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

Then, I would like to thank the colleagues, and co-authors, met during my visiting period at UZH - in particular, my supervisor at UZH, Claudio J. Tessone: thank you for your patient support and for the opportunity I was given to prolong my research period.

Naturally, I would also like to thank my colleagues at IMT, for their invaluable friendship throughout my studies.

Finally, I would like to thank my relatives for their wise counsels and sympathetic ear: they are always there for me.

Vita

June 4, 1991	Born in Rome (Italy)
2016-2021	PhD in Economics, Management and Data Science IMT School for Advanced Studies Lucca (Italy)
2013-2016	Master Degree in Applied Mathematics Final mark: 110/110 cum laude University of Rome 'La Sapienza' (Italy)
2010-2013	Bachelor Degree in Mathematics Final mark: 110/110 cum laude University of Rome 'La Sapienza' (Italy)
2019	Erasmus program, University of Zürich (Switzerland)
2015	Erasmus program, Queen Mary University of London (United Kingdom)

Publications

1. **Exploring the Bitcoin mesoscale structure**
   NICOLÒ VALLARANO, CLAUDIO J. TESSONE, TIZIANO SQUARTINI
   in preparation (2021)

2. **Fast and scalable likelihood maximization for Exponential Random Graph Models with local constraints**
   NICOLÒ VALLARANO, MATTEO BRUNO, EMILIANO MARCHESE, GIUSEPPE TRAPANI, FABIO SARACCO, GIULIO CIMINI, MARIO ZANON, TIZIANO SQUARTINI
   arxiv:2101.12625 (accepted for publication on Scientific Reports) (2021)

3. **Bitcoin Transaction Networks: an overview of recent results**
   NICOLÒ VALLARANO, CLAUDIO J. TESSONE, TIZIANO SQUARTINI
   Front. Phys. **8**, 286 (2020), DOI: 10.3389/fphy.2020.00286

4. **The evolving liaisons between the transaction networks of Bitcoin and its price dynamics**
   ALEXANDRE BOVET, CARLO CAMPAJOLA, FRANCESCO MOTTES, VALERIO RESTOCCHI, NICOLÒ VALLARANO, TIZIANO SQUARTINI, CLAUDIO J. TESSONE
   arxiv:1907.03577 (under submission) (2019)

5. **Network-based indicators of Bitcoin bubbles**
   ALEXANDRE BOVET, CARLO CAMPAJOLA, JORGE F. LAZO, FRANCESCO MOTTES, IACOPO POZZANA, VALERIO RESTOCCHI, PIETRO SAGGESE, NICOLÒ VALLARANO, TIZIANO SQUARTINI, CLAUDIO J. TESSONE
   arxiv:1805.04460 (under submission) (2018)

Presentations

1. **NETWORKS 2021 - A Joint Sunbelt and NetSci Conference (online, 2021)**
   Fast and scalable likelihood maximization for Exponential Random Graph Models (under submission)
   NICOLÒ VALLARANO, MATTEO BRUNO, EMILIANO MARCHESE, GIUSEPPE TRAPANI, FABIO SARACCO, GIULIO CIMINI, MARIO ZANON, TIZIANO SQUARTINI
2. **COMPLENET 2020 -** 11^{th} **International Conference on Complex Networks (Exeter, 2020)**
   A network analysis of Bitcoin transactions
   NICOLÒ VALLARANO, TIZIANO SQUARTINI, CLAUDIO J. TESSONE
3. **COMPLEX NETWORKS 2019 -** 9^{th} **International Conference on Complex Networks and their Applications (Lisbon, 2019)**
   A network approach to the analysis of Bitcoin
   ALEXANDRE BOVET, CARLO CAMPAJOLA, FRANCESCO MOTTES, VALERIO RESTOCCHI, NICOLÒ VALLARANO, TIZIANO SQUARTINI, CLAUDIO J. TESSONE

Abstract

The topic of the thesis is the analysis of the most popular cryptocurrency from a network perspective. Specifically, it focuses on a bunch of network representations of the Bitcoin digital transactions that are studied at different scales (*micro*, *meso* and *macro*) by employing tools from physics, statistics and economics. The thesis is divided into five chapters. Chapter 1 introduces the broad field of cryptocurrencies. Chapters 2 and 3 are dedicated to the understanding of the network properties of the Bitcoin cryptocurrency from both a static and a dynamical perspective and to the investigation of the relationships between the latter ones and purely financial quantities like the BTC price. Chapter 4 is dedicated to the study of the Bitcoin Lightning Network, a recently-developed protocol to speed up the blockhain-based payment system Bitcoin rests upon. Chapter 5 illustrates the substantial modifications that have been required by state-of-the-art algorithms to solve null models for networks on very large networks - as the ones characterizing Bitcoin throughout its entire history.

Chapter 1

Prologue

Nanos gigantium humeris insidentes.

Anonymous

Todo lo que empieza como comedia acaba como ejercicio criptográfico.

Roberto Bolaño

In the fall of 2018 a friend of mine - who is going to remain anonymous, not just pseudo-anonymous! - told me to invest into cryptocurrencies. At the time, I was aware of the Internet new shiny toy but, either for lack of money or for lack of will, I had not invested before.

Anyway, this time was different: for the first time in my life I had a disposable income and was eager to lose it. As it was foreseeable[1], we invested and we lost but we learnt something, i.e. that there is a lot to learn. The crypto-world was the far-west (it is still kind of) and, in some sense - someone, here, would say the truest sense - we were explorers. In fact, what is a researcher if not an explorer? Well, if this metaphore is true, the present thesis is our travel log.

[1] Eventually we won: most of the interesting things happen in the short run but the significant ones tend to play the long game.

In the following hundred pages, we will mainly discuss the nature of Bitcoin Transaction Networks (hereby, BTNs), i.e. the networks of transactions in bitcoins. A transaction is a bilateral relation of money exchange between two alphanumeric addresses (representing the starting and the ending point of the money flow) and represents the basic element recorded on the blockchain.

Bitcoin Transaction Networks pose a set of non-trivial challenges:

- they are awfully large;
- there are many of them - surely, as many as you want;
- the information they convey is noisy.

BTNs are large. As Bitcoin has gained popularity, the number of daily transactions has increased as well; as a consequence, these networks have reached a size of (the order of) millions of nodes. While being very sparse, they still represent a computational challenge for most of the algorithms constituting the (actual) toolbox for network analysis. This was the first challenge we faced: as a consequence, an entire chapter of the thesis is dedicated to the development of alternative algorithms, specifically designed to work on huge, sparse networks.

BTNs are a lot. Bitcoin Transaction Networks are built from transactions, as we said: this introduces time as a fundamental factor as researchers must choose the right time-frame to collect and aggregate transactions. While there are some natural choices, the optimal one is still debated; in general, it depends on the actual phenomenon researchers want to study. A consequence of the presence of time is that, depending on how refined the chosen temporal partition is, the number of Bitcoin Transaction Networks to study increases. Naturally, there is a trade-off between the time-frame of aggregation, the size of the networks and the total number of networks to study: still, the computational effort remains large.

BTNs are noisy. Last but not least, Bitcoin Transaction Networks convey noisy information. Ideally, we know that each transaction in bitcoins happens between two users, be they business firms or private individuals. Yet, what we can observe on the blockchain[2] is not a collection of users exchanging money: what we *actually* observe is money being passed. Imagine you were able to obtain a picture of physical exchange of money - literally, coins and banknotes - in Italy, across one day but the only thing you can appreciate in the picture, about the people exchanging money, is their hands. Millions of hands exchanging money. At a first sight this is how our public record appears: lots of transactions, with little information on the actual actors behind the transactions themselves. Luckily, the mental experiment above is not a good representation of reality because we have some methods to infer the identity of Bitcoin users - and we are going to describe them.

Why did we chose to study Bitcoin Transaction Networks, then? Well, not so many people have studied them before and we thought we could add something to the field. On a more philosophical level, we started by asking ourselves whether there was a relationship between endogenous and exogenous factors in the Bitcoin ecosystem; in other words, if there were a relationship between the network properties of bitcoin transactions and the bitcoin dollar price: *could we describe the price evolution by just observing the network evolution?* We got interested in the peculiarities of crypto-markets and tried to test whether these same ideas were reflected in the topological structure of the underlying networks.

The result of the questions we asked and the answers we found, of the methods we used and the ones we developed is the content of the present thesis. We will start presenting some of the basic concepts needed to understand the ecosystem of cryptocurrencies, together with some mathematical formalization that will come handy later on. Then, a detailed analysis of the transactions data for Bitcoin will follow, at the micro-, meso- and macro-scopic level, in order to detect the peculiarities of the Bitcoin Transaction Networks and of their evolution across time. Last,

[2]The blockchain is a public record of transactions.

we'll dive into some theoretical methodologies, refining and proving some novel techniques to reconstruct large networks from partial information. While the last chapter refers, more generally, to the theory of statistical mechanics of networks, the methodologies presented there are applied to Bitcoin Transaction Networks, thus motivating its presence in this thesis.

Chapter 2

A quick intro to cryptocurrencies

This chapter provides a broad overview of the topic of cryptocurrencies: the history of the idea of constructing a fully-decentralized tool for economic payments is briefly reviewed and the main features of the Bitcoin ecosystem are described. Moreover, we explain how the so-called Bitcoin Transaction Networks have been defined from the data downloaded from the blockchain.

2.1 In principle was Satoshi Nakamoto

It all started with a *white paper*[1], appeared online in October 2008. The unknown Internet persona Satoshi Nakamoto published a sort of report (see Nakamoto (2019)) where the limitations of the online payment systems, at the time, were outlined and a technical solution to overcome them was proposed.

At the time Satoshi Nakamoto was writing, the dominant paradigm for online payments required the presence of a *third party*. Third parties ensure the trustworthiness of the two unknown partners involved in a transaction: on the one hand, they collect and secure (ideally) private

[1] A white paper is commonly considered part of the so-called *grey literature*, i.e. materials produced outside the traditional academic channels.

information on the partners themselves; on the other, they play the role of referees in disputes. As it can be easily imagined, the role of third parties is often played by financial institutions.

Nakamoto, however, shows how the presence of third parties puts a series of undesired constraints on online payments:

- third parties require fees to operate, thus adding economic burden on each transaction;
- third parties make disputes possible, hence making online payments reversible, while the underlying services often are not;
- information disclosure required by the trust-based model may lead to a security breach in partners' privacy as data storage exposes itself to hacking.

The new digital currency Satoshi Nakamoto proposes, i.e. *bitcoin*, 'would be the electronic equivalent of cash' (see Nakamoto (2019)), hence reproducing its three main features:

- cash is free from *third-party fees*: I don't have to spend additional money only to execute a transaction, while mainstream forms of electronic payments requires a price (i.e. the user fee) to validate the transaction. The fee is often (but not exclusively) on the merchant side;
- cash is *non-reversible*: when I pay the groceries with cash, the transaction is *de facto* non-reversible; once I have paid, there is no mechanism embodied in my money which can fly it back to me, had I a claim on the quality of the food I have just bought[2];
- cash is *anonymous*.

[2]Naturally, I can always recourse to law to solve disputes but police does not oversees each economic transaction I execute.

2.2 Bitcoin and before

The idea of electronic payments arose with the birth of Internet: already in the 1980s, Chaun proposed some guidelines for a 'blind signature technology' based on cryptographic tools (see Chaum (1983)). By reading Chaun's manuscript we realize how the issues of privacy, security and control were already intertwined in the mind of theorists. In the following years, many alternative digital currencies were proposed, e.g. universal electronic cash (see Okamoto et al. (1991)), untraceable off-line cash (see Brands (1993)), fair blind signatures (see Stadler et al. (1995)), etc.

Some of the ideas influencing later developers were already present during the embryonic stage: for example, B-money (see Dai (1998)) for the first time proposed to solve a computationally intense puzzle for mining. All proposed solutions, however, ultimately failed either to provide a working technology for decentralised transactions or simply to gain public attention: as a consequence, the only standard for online payments emerged from the Nineties was the centralised one, based upon the presence of third-parties (e.g. Paypal and other bank-related services).

The Bitcoin ecosystem was announced by Satoshi Nakamoto in 2008 and deployed in 2009[3]. What is Bitcoin? This obscure term rose to worldwide recognition in recent years, making its way trough enthusiastic blogs and scientific papers to reach the front pages of newspapers all over the world. Several definitions have been proposed, e.g. 'a chain of digital signatures' (see Lischke et al. (2016) and Singh et al. (2021)), 'a distributed, public ledger that contains the history of every bitcoin transaction' (see Dwivedi et al. (2019) and Singh et al. (2021)), 'a digital ledger in which transactions made in bitcoin or another cryptocurrency

[3] After the success of Bitcoin, many alternative electronic coins have been created in the last years - informally named *altcoins*: the website `www.coinmarketcao.com` has estimated that there are more than 7.000 cryptocurrencies at the time of the writing, i.e. 2020; while many of these are just Bitcoin clones, some genuinely innovative ideas providing different solutions from the Bitcoin one are, however, present: examples are provided by Ethereum, Litecoin, Monero, etc.

are recorded chronologically and publicly' (see Singh et al. (2021)) or 'a decentralized database containing sequential, cryptographically linked blocks of digitally signed asset transactions, governed by a consensus model' (see Singh et al. (2021) and Sultan et al. (2018)). As it can be read in Antonopoulos (2014): 'Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem'.

To fully understand the last sentence, let us think about the old, good, reliable physical money: is cash just the paper it is printed on? Money is, first of all, an idea based on trust - about its value and the issuing institution - and only afterwards it is also a physical 'infrastructure' (i.e. the paper it is printed on). So once again, what is Bitcoin? Bitcoin is a digital currency whose units are called bitcoins[4]: the latter ones are used to store and exchange values (between the participants of the Bitcoin network).

2.3 The blockchain technology

Let us now list the keywords referring to the Bitcoin environment and describe in detail the related concepts.

Transactions. Transactions are the foundation stones of the Bitcoin ecosystem. They are nothing else than a collection of *inputs*, i.e. *debts* towards a Bitcoin account, and a collection of *outputs*, i.e. *credits* towards another Bitcoin account. Usually, the total sum of the outputs is a bit less than the total sum of the inputs: the difference is the *transaction fee*, a sort of 'reward' collected by the *miner* who is going to add the transaction to the *block* that will become part of the ledger[5]. Looking at figure 1, we observe that inputs and outputs are referred to as to alphanumeric strings: these are 'signatures', often called *addresses*, that certify the ownership of the bitcoins contained in the inputs. From a theoretical point of view, a transaction is a data structure containing the following information:

[4]Throughout this work, the word *bitcoin* will be employed to indicate the currency, the word *BTC* will be employed as an abbreviation of the latter one and the word *Bitcoin* will be employed to indicate the entire system.

[5]Transaction fees are conceptually different from the third-party fees Nakamoto wanted to get rid of: they prevent denial-of-service attacks and incentivize miners.

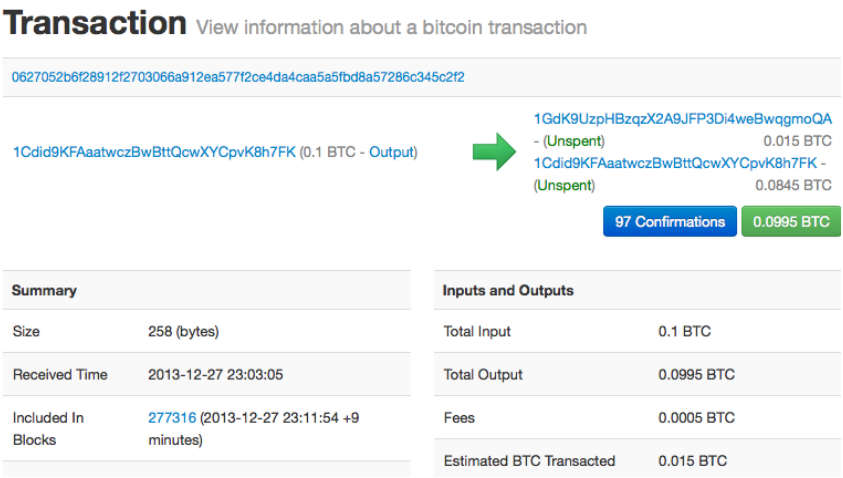

Figure 1: The way a transaction appears to an online blockchain explorer. The transaction is already registered on the blockchain (you can spot this from the 97 confirmations recorded). Figure from Antonopoulos (2014).

- *version*: it specifies the rules followed by any transaction - in fact, the Bitcoin version may change over time;
- *input counter*: it specifies the number of inputs;
- *inputs*: one or more input data structures. They point to unspent transaction outputs and to the unlocking scripts (the keys) to spend them;
- *output counter*: it specifies the number of outputs;
- *outputs*: one or more output data structures. They contain the amount of bitcoins received and the locking script to claim them;
- *locktime*: a Unix time-stamp, also known as *block-height*. It specifies the time the transaction should be added to the blockchain[6].

[6]The way locktime works is similar to the way post-dated payments work.

Keys. Users have cryptographic keys that allow them to 1) claim the ownership of their bitcoins and 2) spend them. As emerged from the description of transactions, inputs are the outputs of previous transactions: the way a new input actually claims bitcoins from an unspent output (inter-locking the two transactions) is via the keys owned by the user. Whenever a user wants to issue a payment, he needs to write down a transaction on the blockchain; to do so, he must prove that he owns the money he wants to spend. Claiming ownership on unspent bitcoins means providing the script which unlocks the counterpart locking script (i.e. that of the unspent transactions the user is claiming). The *unlocking script* takes the user's cryptographic and returns back the solution to the locking script.

As the unlocking script is able to satisfy the locking script only trough a specific key, we say that the key provides the user *ownership* of the unspent output value. As bitcoins move from address to address - and, therefore, from user to user - the sequence of inputs and outputs creates a chain of 'ownerships': this example should clarify that the concept of 'ownership' in Bitcoin coincides with the concept of 'control' of the keys of unspent outputs on the public ledger[7].

The wallet. Each user collects all his keys via a software called *wallet*. To be noticed that, in our research, we will identify a Bitcoin user with the wallet he controls, i.e. with the set of blockchain inputs he is able to move. The first task of the wallet software is that of constructing well-defined transactions: once a transaction is constructed, it needs to be verified and stored. The second task of the wallet software is that of broadcasting it on the Bitcoin peer-to-peer network: within seconds, all nodes are notified about the transaction. Afterwards, the transaction is verified and recorded on the blockchain, a process known with the name of *mining*.

[7]In real life, if I have 50 $, the ownership of this money is certified by the physical possession of the bill. In the Bitcoin ecosystem, instead, I posses 50 bitcoins if I 'posses' the corresponding cryptographic key.

The mining mechanism. From the pool of unverified transactions a bunch of transactions is pulled together into a *block*. In order to prove a block, users called *miners* must solve a computationally-intense problem which is dynamically adjusted to require, on average, 10 minutes to be solved, regardless of the numbers of miners and their computational power. This process is called *proof-of-work* (see Nakamoto (2019)): once the problem has been solved, the miner is awarded with bitcoins which become part of the currency pool.

Quoting Antonopoulos (2014): 'A good way to describe mining is like a giant competitive game of Sudoku that resets every time someone finds a solution and whose difficulty automatically adjusts so that it takes approximately 10 minutes to find a solution. Imagine a giant Sudoku puzzle, several thousand rows and columns in size. If I show you a completed puzzle you can verify it quite quickly. If it is empty, however, it takes a lot of work to be solved! The difficulty of the Sudoku can be adjusted by changing its size (more or fewer rows and columns), but it can still be verified quite easily even if it is very large. The puzzle used in Bitcoin is based on a cryptographic hash and exhibits similar characteristics: it is asymmetrically hard to solve, but easy to verify and its difficulty can be adjusted'.

Verifying the validity of a block of transactions is computationally trivial - as much as it would be for a computer script to verify the validity of a completed Sudoku puzzle. Once a miner has solved the problem, he broadcasts the solution to the Bitcoin network: verified blocks are the ones on whose reality users reached a consensus. Once the blocks are verified, they are chained in chronological order: this ensures that older blocks become 'safer' with time since the are proved both in a directed and in an undirected fashion (i.e. via the proof of the subsequent blocks).

Beside solving the verification problem of transactions in a fully decentralized fashion, the mining mechanism also solves the coin-issuance one. In fact, once a block becomes part of the blockchain, it also contains a new transaction that transfers newly minted bitcoins to the miner who proved the block[8]. Hence, miners are incentivized by the so-called

[8] As of 2020, it amounts at 12.5 bitcoins.

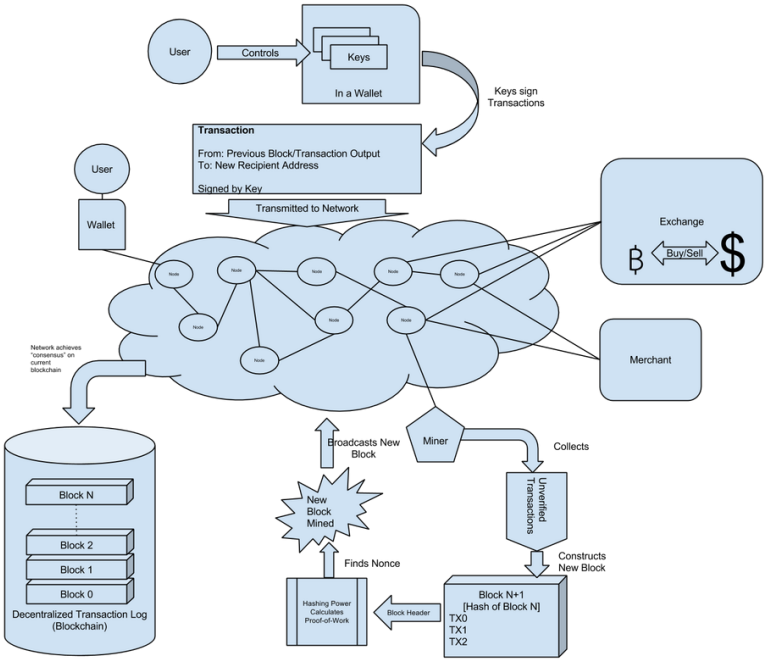

Figure 2: Pictorial representation of the Bitcoin ecosystem. Figure from Antonopoulos (2014).

transaction fees.

The blockchain. Any new block of verified transactions is written on the blockchain: the new block is received by all the nodes of the network and becomes part of the official, *publicly available*, transaction ledger. All transactions among users are stored there.

The Bitcoin protocol. It is the backbone of the *peer-to-peer network* employed by Bitcoin users to communicate among them over the Internet. Each Bitcoin user, i.e. anyone who runs a stack of the Bitcoin protocol on his computer, stores a copy of the blockchain (i.e. a copy of all verified transactions) locally.

2.4 From pseudo-anonymity to users

What do we mean by *pseudo-anonymity* in Bitcoin? As we mentioned before, users execute payments trough addresses. More specifically, an address is the double ash of a public key derived from an ECSDA key pair (see Antonopoulos (2014)). Since creating addresses comes at no cost, the rule of thumb is that of creating a new address for each operation. Thus, when we speak of pseudo-anonymity, we mean that each address is an alias of a user - without revealing any information about the person or the other addresses he/she may control. This is the only anonymity protection hard-coded in Bitcoin[9].

Although Bitcoin anonymity is far from being unbreakable, when it was implemented it was the best alternative to the mainstream electronic payments, thus creating a viable option for those subjects interested in moving money on the Internet without revealing their identity. Driven by the needs of law-enforcement agencies, digital-economics researchers and industrial espionage, the activity in the field of Bitcoin de-anonymization has grown over the years. Last but not least, knowing the limits of the anonymity provided by the system is also of interest for the Bitcoin users themselves.

As researchers, our interest in the Bitcoin de-anonymization field lies in the possibility to retrieve, within a certain degree of precision, the actual users behind the overwhelming amount of addresses we actually see on the blockchain. Naturally, we are not interested to identify individuals in real life but spotting the economic subjects of transactions may transform an interesting - yet very noisy - data set as the blockchain into an economically meaningful snapshot of the Bitcoin landscape.

To this aim, several *heuristics* can be employed - a 'heuristic' being a set of rules that take advantage of the Bitcoin protocol to identify owners of different addresses (see Androulaki et al. (2013), Tasca et al. (2018), Harrigan et al. (2016), and Ron et al. (2013)). A brief description of the two most common ones follows.

[9]Other methods to protect identity have been invented through the years, e.g. the Bitcoin mixers.

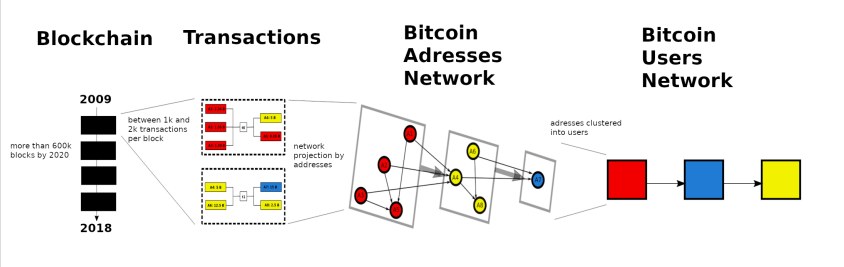

Figure 3: From left to right: the blockchain as a sequence of clustered transactions; two transactions stored in the blockchain; our first network representation, i.e. the Bitcoin Address Network; our second network representation, i.e. the Bitcoin User Network. The figure also shows how the heuristics work: coloured boxes/nodes represent ownership of the same user.

Multi-input heuristics. This heuristic is generally believed to be the safest one for clustering addresses and is based on the assumption that, if two (or more) addresses are part of the input of the same transaction, they are controlled by the same user. The key idea behind this heuristics is that, in order to produce a transaction, the private keys of all addresses must be accessible to the creator of the transaction.

Change-address identification heuristics. Transaction outputs must be fully spent upon re-utilisation. Hence, the transaction creator usually controls also one of the output addresses. Specifically, if an output address appears for the first time and the amount transferred to it is lower than all the inputs, then it is likely to belong to the input user.

2.5 An overview of the Bitcoin price evolution

Bitcoin price has undergone an interesting evolution, with periods of intense growth followed by sudden and large decreases. In the literature, these periods are known as *bubbles*, a term that indicates an unsustainable price growth (e.g. of a good) since not justified by its underlying

Bubble	Start	End	Days	Growth	Mean ret.
1	25-05-2012	18-08-2012	84	3.1	0.013
2	03-01-2013	11-04-2013	98	20.4	0.031
3	07-10-2013	23-11-2013	47	6.8	0.042
4	08-06-2015	18-12-2017	924	103	0.005
5	31-03-2017	18-12-2017	155	21	0.02

Table 1: Time intervals of the four main bubbles occurring between May 2012 and December 2017 (see Wheatley, Spencer et al. (2018)). Bubble 5 overlaps with the last six months of Bubble 4.

value. As it can be easily imagined, the literature on Bitcoin financial bubbles is strictly related to the study of the Bitcoin fundamental value: we refer the interested reader to Bergstra et al. (2014), Yermack (2015), Van Alstyne (2014), Bouoiyour et al. (2015), and Fantazzini et al. (2017) for the discussion about the Bitcoin value and to Garcia et al. (2014) and Wu, Ke et al. (2018) for the discussion about price modeling.

In the present thesis we will discuss the bubble detection method proposed in Wheatley, Spencer et al. (2018). The authors of the paper combine two different approaches:

- *Metcalfe law*, stating that the value of a network is proportional to the square of its number of nodes (see Metcalfe (2013));
- *Log-Periodic Power-Law Singularity Model (LPPLSM)* that identifies two empirical features of bubbles: faster-than-exponential growth and accelerating log-periodic volatility fluctuations (see Sornette et al. (2014)).

By combining the Metcalfe law to proxy the Bitcoin fundamental value and the LPPLSM as a technical measure to diagnose financial bubbles, the authors are able to indicate periods of impending bubbles and subsequent crashes. More detailedly, they identify four main bubbles, reported in table 1. Interestingly enough, the authors also identify triggering moments of bursting/crashing price for the dates reported in table 1:

- 19 June 2011: Mt. Gox is hacked, causing the Bitcoin price to fall of 88% over the next three months;

- 28 August 2012: Ponzi fraud of hundreds of thousands of bitcoins under the name 'Bitcoins Savings and Trust' (charges filed by the Securities and Exchange Commission);

- 10 April 2013: the major Bitcoin exchange Mt. Gox breaks under high trading volume. The price falls more than 50% over the next two days;

- 05 December 2013: the People's Bank of China bans financial institutions from using Bitcoin. Bitcoin market cap drops of 50% over the next two weeks;

- 07 February 2014: operational issues at major exchanges due to distributed denial-of-service attacks - two weeks later Mt. Gox closes;

- 28 December 2017: reports that South Korean regulators threatened to shut down cryptocurrency exchanges.

2.6 An overview of networks

Let us now briefly recall what a network is. A *network* (or *graph*) is a set or *node* (or *vertices*) connected by *links* (or *edges*). When links represent relationships between node whose specification does not need a direction, the underlying network is said to be *undirected*; otherwise, when one needs to assign a direction to the links, we speak of *directed* networks. It is also possible to assign a *weight* to the links: in this case, one speaks of *weighted* networks. In this thesis, we will consider neither *self-loops* (i.e. edges connecting a node with itself) nor *multiple edges* (nodes connected by more than one edge) and focus on binary[10], undirected and directed representations only.

[10]The information carried by weights is of great interest but would have increased the complexity of the whole data analysis procedure too much.

While the abstract notion of network is a purely mathematical concept, there are many ways of representing a graph. The commonest one is based on the definition of the *adjacency matrix*: given the graph $G = (V, E)$ (i.e. the pair of node set and link set), the corresponding adjacency matrix $\mathbf{A}$ is an $N \times N$ square matrix, with $|V| = N$ whose generic entry a_{ij} can be defined as follows. Let us now consider two generic nodes, i and j: if a link *between*[11] i *and* j exists, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. Undirected networks are represented by symmetric adjacency matrices; directed networks, in general, are not.

2.7 The Bitcoin Transaction Networks

Bitcoin data were collected from the Bitcoin public ledger, from 5th January 2009 to 25th June 2018. More specifically, our data consist of $304.111.529$ addresses among which a total number of $283.028.575$ transactions take place. Our heuristics allowed us to identify $16.749.939$ users, among which $224.620.265$ transactions were found to take place. In terms of traded volume, the transactions between users and addresses amount at $3.114.359.679$ and $4.432.597.496$ bitcoins respectively.

The apparent inconsistency between users' and addresses' volumes of exchanged bitcoins has an easy explanation. Transactions among addresses controlled by the same users do not add up to the total volume of bitcoins exchanged when addresses are clustered in users: they just disappear because they correspond to the activity of someone moving money from one of his bank accounts to the others. On the contrary, when we focus on the address representation these 'fake' transactions add up to the volume - thus, explaining the larger volume observed in this second case.

Let us now describe how we obtained our sequence of Bitcoin Transaction Networks. First, we have fixed a time-span[12] Δt. Then, we have

[11] In this example, we consider an undirected network. In case it were directed, the link would be *from* i *to* j.

[12] See later for a discussion about the temporal intervals employed for this analysis.

split the whole history of Bitcoin in intervals of length Δt. For each interval, we have defined a BAN, i.e. a Bitcoin Address Network (in symbols, $\mathbf{A}_{(t)}^{\text{BAN}}$) and a BUN, i.e. a Bitcoin User Network (in symbols, $\mathbf{A}_{(t)}^{\text{BUN}}$).

The first representation has been obtained quite straightforwardly: each transaction registered on the blockchain, during interval t, has been treated as a link between the nodes (i.e. the addresses) involved in the transaction itself: hence, the value of the corresponding entry of the adjacency matrix $\mathbf{A}_{(t)}^{\text{BAN}}$ has been set to 1. Naturally, the link direction is induced by the 'status' of the nodes (either input or output) in the transaction.

Given the whole set of addresses, we have applied a combination of the two aforementioned heuristics to cluster them into users. Starting from the network of addresses, where any two of them are connected if participating to the same transaction, we have, first, applied the multi-input heuristics and clustered any two addresses together if both were found to participate to a given transaction as inputs. This led us to a network of 'intermediate' users. Afterwards, we have employed the second heuristics and 'assigned' the change address of each transaction to the input user (i.e. to the set of addresses appearing as inputs and clustered together into the same 'intermediate' user).

Then, for each interval t, we defined a BUN, i.e. a Bitcoin User Network (in symbols, $\mathbf{A}_{(t)}^{\text{BUN}}$) starting, once again, from the transactions registered on the blockchain, during interval t: now, the role of nodes is played by the users that took part in the transactions themselves - the latter ones, acting as links.

Generally speaking, the BANs and the BUNs are directed, binary graphs that satisfy these properties:

- BANs: neither the identity of nodes, nor their number is constant across the entire Bitcoin history (i.e. node 1 on snapshot i may not coincide with node 1 at later times);
- BUNs: as for the BANs, neither the identity of nodes, nor their

number is constant across the entire Bitcoin history;

- while addresses have a one-use policy, users entities supposedly appear more than once across the entire Bitcoin history;
- the number of users is, by construction, strictly less than the number of addresses, in turn implying that BUNs are computationally easier to handle;
- BUNs are supposed to provide a picture of the Bitcoin history less affected by noise, since clusters of addresses proxy the behaviour of actual users better than plain addresses.

We would also like to stress that, while considering *only* the address-based representation may be misleading, given the potentially large number of different addresses controlled by the same users, relying *only* on the user-based one requires a very accurate clustering procedure, as pointed out in Di Francesco Maesa et al. (2019), Fergal et al. (2013), and Harrigan et al. (2016); for these reasons, whereas possible, we have opted to carry out a comparative analysis of the two.

Let us now comment on the choice of the time-span Δt employed in the present analysis. Here, Δt is both a *day*-lasting and a *week*-lasting interval. The time interval is measured on the blockchain at the block level: since each block has a timestamp to record the exact time it was mined, we are able to aggregate transactions at the desired level of (temporal) detail by playing with it.

2.8 The Bitcoin Lightning Network

Bitcoin is affected by the so-called *scalability problem*. In other words, only a limited number of transactions per second can be served by the Bitcoin network: in December 2020 the rate of processed transactions amounted at $\simeq$ 2.000 every 10 minutes[13] - a ridiculous number when compared

[13] `https://www.blockchain.com/charts/n-transactions-per-block`

to the performance of traditional, centralized online payment networks, that are able to verify thousands of transactions per second. The increase of Bitcoin popularity made the scalability problem only more evident.

Proposed in 2015 (see Poon et al. (2016)), the *Bitcoin Lightning Network* (BLN) is a 'Layer 2' protocol that can operate on top of blockchain-based cryptocurrencies by creating bilateral channels for off-chain payments which are, then, settled on the blockchain, once the channels are closed. As both the transaction fees and the blockchain confirmation are no longer required, the network is spared from avoidable burden.

The BLN is constructed in a fashion that is similar to the way the BAN is defined: it is an undirected, weighted graph whose nodes are the addresses exchanging bitcoins on the 'Layer 2'. Formally, for each time step t, the BLN is represented by a weighted adjacency matrix $\mathbf{W}_{(t)}^{\text{BLN}} \in \mathbb{R}^{N^{(t)} \times N^{(t)}}$ where $N^{(t)}$ is the number of nodes at step t. The generic entry $w_{ij}^{(t)}$ represents the total amount of bitcoins exchanged between nodes i and j during the time interval t. The binary adjacency matrix $\mathbf{A}_{(t)}^{\text{BLN}}$, instead, is defined as the binary projection of $\mathbf{W}_{(t)}^{\text{BLN}}$, constructed following the rule $a_{ij}^{(t)} = \mathbb{1}_{[w_{ij}^{(t)} > 0]}$.

As for the BANs and the BUNs, the choice of the time partition defines the collections of networks under analysis. Three different representations of the BLN were considered in the present work, i.e. the daily one, the weekly one and the daily-block one: while a daily/weekly snapshot includes all channels that were found to be active during that day/week, a daily-block snapshot consists of all channels that were found to be active at the time the first block of the day was released (hence, the transactions considered for the daily-block representation are a subset of the ones constituting the daily representation). The BLN was considered across a period of 18 months, i.e. from 14th January 2018 to 13th July 2019, at the end of which the network consisted of 8.216 users, 122.517 active channels and 2.732,5 transacted bitcoins (Lin et al. (2020)).

The extent of the analysis whose results constitute the present thesis

is unprecedented under several respects: to the best of our knowledge, in fact, published contributions typically focus on 1) a (much) shorter time span and 2) a single Bitcoin representation, i.e. either the address-based one (see Kondor et al. (2014) and Popuri et al. (2016)) or the user-based one (see Androulaki et al. (2013), Di Francesco Maesa et al. (2016a), Di Francesco Maesa et al. (2016b), Di Francesco Maesa et al. (2018a), Di Francesco Maesa et al. (2017), Di Francesco Maesa et al. (2018b), Di Francesco Maesa et al. (2019), Javarone et al. (2018), Lischke et al. (2016), Meiklejohn et al. (2013), Ober et al. (2013), Reid et al. (2013), and Ron et al. (2013)).

Chapter 3

Bitcoin at the microscale

The content of this chapter overlaps with the one of the papers authored by Bovet et al. (2019) and Vallarano et al. (2020). It focuses on the description of the local structure of our Bitcoin Transaction Networks (from both a static and a dynamical perspective) and investigates the presence of relationships between the latter ones and purely financial indicators as the BTC price.

3.1 Bitcoin Transaction Networks: an overview

During the nine years under analysis, Bitcoin adoption rose worldwide. This is clearly reflected in the evolution of the total number of nodes N (be they addresses or users) and links L, as showed in figure 4: both steadily increase over time, irrespectively from the specific representation considered (i.e. BAN_{day}, BAN_{week}, BUN_{day}, BUN_{week}).

On the other hand, the link density (or connectivity)

$$d = \frac{L}{N(N-1)} = \frac{\sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} a_{ij}}{N(N-1)} \tag{3.1}$$

decreases: this implies that the total number of links does not increase proportionally to the square of the number of nodes, as per equation (3.1); equivalently, we can say that the (average) number of transactions per user, i.e. $\overline{k} = \frac{L}{N}$, does not increase proportionally to N, i.e. Bit-

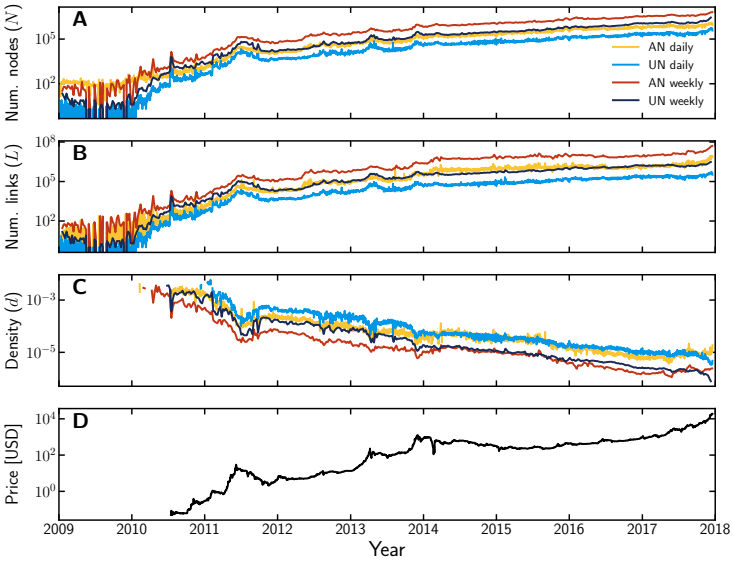

Figure 4: Evolution of basic statistics for the four Bitcoin network representations considered here (BANs and BUNs on a weekly and a daily basis): (A) number of nodes N, (B) number of links L and (C) link density d (notice that the link density is computed for networks with at least 500 nodes). The fourth panel (D) shows the evolution of the Bitcoin price in USD (since when trading bitcoins for USD has started happening on a more regular basis). Figure from Bovet et al. (2019).

coin users engage in a 'finite'[1] number of transactions; other real-world economic and financial networks are, instead, observed for which the average degree scales with the size of the network, the most prominent example being the World Trade Web, whose connectivity reads $c_{\text{WTW}} \simeq 0.5$ throughout its entire history. Equivalently, one can say that the total number of links grows linearly with the total number of nodes, i.e. $L = O(N)$ (see Aspembitova et al. (2019)) - hence, $d = O(N^{-1})$.

[1] Actually, quite low as we will see later.

For the sake of comparison, we have also plotted the evolution of the Bitcoin price in US dollars (USD, bottom panel in figure 4): while the price reached its peak in December 2018, the history of the Bitcoin pricing is a troubled one, with many ups and downs.

3.2 Bitcoin Transaction Networks: degree distributions

The Bitcoin Transaction Networks we consider here are binary, directed networks. The simplest, yet non-trivial, quantity to monitor is the number of neighbors of nodes, i.e. their *degree*, here indicating the number of transactions a node, be it an address or a user, takes part in - for Bitcoin payments, moving value across wallets, purchasing goods, receiving services, etc. As directed networks are considered, two different notions of degree remain naturally defined, i.e.

Out-degree of node i: the number of outgoing links from node i; in our case, it counts the number of transactions the node i (either a user or an address) participates in, by *sending* bitcoins. It is defined as

$$k_i^{out} = \sum_{i=1}^{N} a_{ij}$$

(where a_{ij} indicates the entry of a generic adjacency matrix $\mathbf{A}$);

In-degree of node i: the number of incoming links towards node i; in our case, it counts the number of transaction node i (either a user or an address) participates in by *receiving* bitcoins. It is defined as

$$k_i^{in} = \sum_{i=1}^{N} a_{ji}$$

(where a_{ij} indicates the entry of a generic adjacency matrix $\mathbf{A}$).

Putting together the in- and the out-degrees for all nodes, we construct the so-called *degree sequences*, defined as $k^{in} = (k_1^{in} \dots k_N^{in})$ and

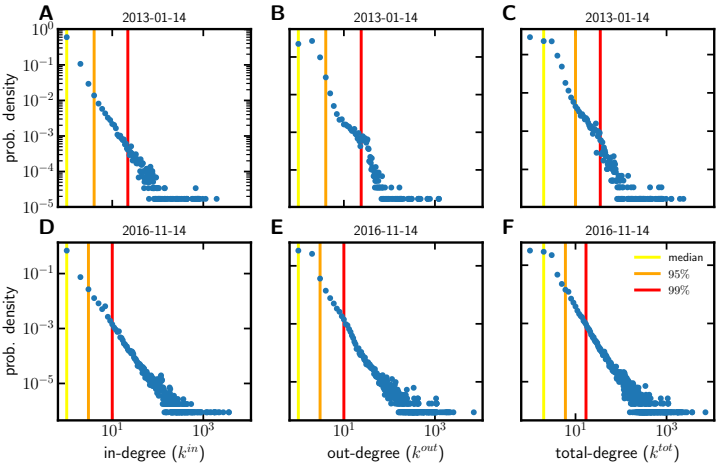

Figure 5: Two weekly snapshots of the BUNs in-degree, out-degree and total degree distributions: the latter ones are heavy-, right-tailed, an evidence suggesting that many nodes with (very) small degree coexist with few large hubs with thousands of incoming and outgoing connections. Figure from Bovet et al. (2019).

$k^{out} = (k_1^{out} \dots k_N^{out})$; one can also define the total degree sequence, as $k^{tot} = (k_1^{tot} \dots k_N^{tot}) = (k_1^{in} + k_1^{out} \dots k_N^{in} + k_N^{out})$. The (empirical) degree distributions induced by the three degree sequences above are shown in figure 5: as we see, these distributions are heavy-, right-tailed, suggesting that many nodes with (very) small degree coexist with few large hubs with thousands of incoming and outgoing connections - in fact, snapshots appear where the degree of the largest hub is one order of magnitude larger than the degree of the second most connected node in the network.

A visual inspection of the degree distributions suggests them to be power-laws. In order to test this hypothesis, we split our time interval in two and employ a double Kolmogorov-Smirnoff test (see Restocchi et al. (2019) and Bauke (2007)), at a significance level of 5%, on each subset.

The date we choose to split our data coincides with 2014-02-24, i.e. the date of the closure of the Mt. Gox exchange market. Interestingly, we find that the hypothesis that the out-degrees are distributed according to a power-law cannot be rejected in 54% of the snapshots, before 2014-02-24; this percentage drops to 26% after that date. For what concerns the in-degrees, the percentage of times that the p-value is larger than 0.05 is 45%, before 2014-02-24 and 60%, after it; for the total degrees, instead, these percentages evolve from 54% to 70%. In conclusion, the failure of Mt. Gox definitely seems to have had an impact on the functional form of the degree distributions.

Although we cannot conclude *tout court* that our degree distributions are power-laws, their heavy-tailedness still suggests that a mechanism similar-in-spirit to the preferential attachment may be shaping our Bitcoin Transaction Networks. This seems to be confirmed by the analysis carried out in Aspembitova et al. (2019) but with an important caveat: a preferential attachment -like mechanism able to 'distinguish' between classes of nodes (in particular, speculators from exchanges) seems to be in place; in any case, the percentage of new users joining Bitcoin to establish a connection with already very-connected nodes (i.e. hubs such as trading places for cryptocurrencies, digital banks converting bitcoins, etc.) can be imagined as not employing bitcoins as a currency but only as an asset of value.

For directed networks, a simple relation holds between the in-degrees, the out-degrees and the total number of links L, i.e.

$$L = \sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} a_{ij} = \sum_{i=1}^{N} k_i^{out} = \sum_{i=1}^{N} k_i^{in} \tag{3.2}$$

as it can easily proven by just swapping the two sums. From equation (3.2) it follows that L is directly proportional to the first moment of both the in- and out-degree distributions. In fact, it is readily seen that $\overline{k} = \frac{L}{N} = \mu[k^{out}] = \mu[k^{in}]$. This allows us to rewrite the link density as

$$d = \frac{\mu[k^{out}]}{N-1} = \frac{\mu[k^{in}]}{N-1}; \tag{3.3}$$

equation (3.3) provides us a further insight on the results of section 3.1: the increasing sparsity of the Bitcoin Transaction Networks shown in figure 4 may be explained by an increase in the number of nodes N potentially accompanied by a decrease of the first moment of the directed degree distributions. To gain further insight into such a behavior let us take a closer look at the evolution of the first moment of the latter ones: as shown in figure 6, panel a), the first moment of the BUNs in- and out-degree distributions, at both the daily and the weekly time scale, is practically constant over the whole period under analysis, i.e. $\mu[k^{out}] \simeq \mu[k^{in}] \simeq O(1)$; this means that the density decrease is 'just' a consequence of the increase of the number of nodes. For what concerns the BANs, a similar conclusion holds, the only difference being that, now, the first moment of both distributions remains finite.

A change in the behavior of the first moment of our BANs in- and out-degree distributions is, however, clearly detectable after 2014: its trend show large variations, bounded in the interval $0 < \mu[k^{out}] \simeq \mu[k^{in}] \lesssim 10$, that persist until the end of the period under analysis. This evidence seems to suggest that a structural change similar to the one observed for the BUNs has taken place for the BANs as well.

Let us now focus on the higher moments of our degree distributions. While the n-th central moment of a random variable x is defined as

$$\mu_n = \mathbb{E}\left[(x-\mu)^n\right], \tag{3.4}$$

with $\mu_1 = 0$ (since $\mathbb{E}[x] = \mu$), μ_2 being the *variance* and $\sigma = \sqrt{\mu_2}$ being the *standard deviation*, its standardized version reads

$$\tilde{\mu}_n = \mathbb{E}\left[\left(\frac{x-\mu}{\sigma}\right)^n\right] \tag{3.5}$$

with $\tilde{\mu}_3$ being the *skewness* and $\tilde{\mu}_4$ being the *kurtosis*. Studying the evolution of the moments of the degree distributions helps to better clarify the

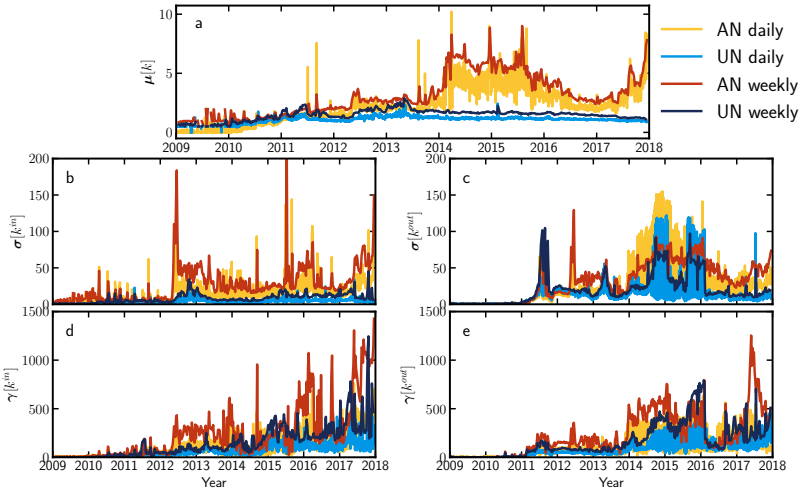

Figure 6: Evolution of the moments of the degree distributions of our BUNs and BANs: a) the average μ; b), c) standard deviation σ; d), e) skewness γ - moments of the in-degrees are shown in panels a), b), c) while moments of the out-degrees are shown in panels a), c), e). While the average degree of the BUNs is practically constant throughout the entire period considered, its trend for the BANs is characterised by peaks and oscillations. Different trends also characterise the evolution of the standard deviation of the in- and the out-degrees: the latter are more heterogeneous than the former ones, especially in the triennium 2014-2016. The behavior of the skewness is, instead, more similar across different representations/time scales. Adapted figure from Bovet et al. (2019).

evolution of the underlying network structure, e.g. by understanding if nodes tend to establish more or less interconnections over time (via the inspection of the first moment), the extent to which the behavior of nodes deviates from the average one (via the inspection of the second moment) and in which direction (via the inspection of the third moment), etc.

Let us now comment on the higher moments evolution (i.e. panels b), e) in figure 6). In-degrees seem to be more homogeneous than out-degrees, especially when considering BUNs: in fact, while the distri-

butions of the out-degrees are characterized by alternating windows of 'low' and 'high' heterogeneity, in-degrees have a steadily smaller standard deviation. For what concerns the BANs, the situation is different: three well-defined peaks, separated by long windows of smaller values, characterize the standard deviation of the in-degrees; on the other hand, the out-degrees standard deviation varies in a smoother fashion. Interestingly, the standard deviation of the BUNs in-degree distributions is always dominated by the standard deviation of the BANs in-degree distributions while this is no longer true when considering out-degrees.

The behavior of the in-degrees skewness is, again, different from that of the out-degrees skewness: while the first one increases over time (with larger peaks characterizing the BANs), this is no longer true for the second one, for which we observe a first 'bump' between 2014 and 2016 and a second one around 2018. The increasing asymmetry of the in-degree distribution may suggest the emergence of large(r) hubs in the Bitcoin ecosystem.

3.3 Network properties versus the Bitcoin price

Let us now analyse the relationship between the evolution of a bunch a network properties and that of the Bitcoin price. First, let us plot the evolution of the total number of nodes and of the link density versus the Bitcoin price (in USD); additionally, let us colour each dot according to the Ratio between current Price and its Moving Average (RPMA) of the Bitcoin price at the time. Given that p_t is the closing price in period t (day or week), the RPMA at time t is defined as

$$\text{RPMA}_t = 100 \log_{10} \left(\frac{p_t}{\frac{1}{\tau} \sum_{s=t-1-\tau}^{t-1} p_s} \right) \tag{3.6}$$

and quantifies the price dynamics: for the weekly Bitcoin Transaction Networks the RPMA has been computed on the previous $\tau = 4$ weeks, while for the daily Bitcoin Transaction Networks it was computed on the previous $\tau = 7$ days. As shown in figure 7, a clear trend appears, indicating that the price and the network size N (the link density d) are, overall,

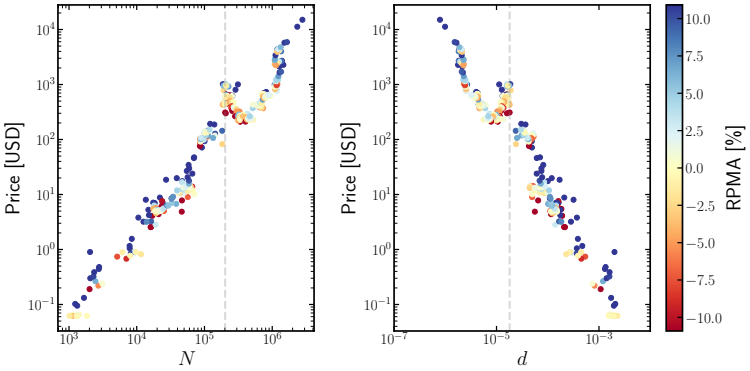

Figure 7: Correlation between the Bitcoin price (in USD) and the basic statistics, i.e. the number of nodes and the link density, for the BUNs at the weekly time scale. Additionally, each dot representing an observation is coloured according to the value of the Ratio between the current Price and its Moving Average (RPMA) indicator. The vertical, dashed line coincides with the bankruptcy of Mt. Gox in February 2014. Figure from Vallarano et al. (2020).

positively (negatively) correlated throughout the entire Bitcoin history. The only exception is represented by the trend inversion starting after the Mt. Gox failure and consequence of the prolonged price decrease observed during the triennium 2014-2016, during which the network size has increased of (almost) one order of magnitude (see also figure 10).

Let us now plot the evolution of the first three (normalized) moments of the out-degree distribution (standard deviation, skeweness and kurtosis) versus the number of nodes of the corresponding snapshot, whereas each dot is coloured as explained above and the dashed line indicates 02-24-0214, i.e. the date when Mt. Gox bankrupted. As figure 8 shows, there is a clear proportionality between the time and the size of the Bitcoin Transaction Networks: while the relationship between the moments and the size looks almost linear before 2014, it is less easy to identify a clear trend, afterwards. While true for both the BANs and the BUNs, this is more evident for the latter than for the former ones - incidentally

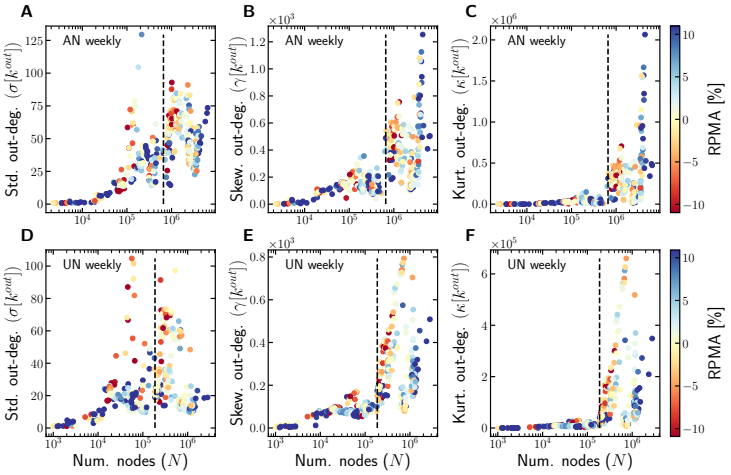

Figure 8: Correlation between the moments of the out-degree distributions, the number of nodes and the RPMA indicator. While the scatter plots depict the relationship between the moments of the out-degree distributions and the number of nodes, each dot is coloured according to the value of the RPMA at that time. The vertical, dashed line coincides with the bankruptcy of Mt. Gox in February 2014. Figure from Bovet et al. (2019).

highlighting the importance of the heuristic-based, data-cleaning step.

Overall, the standard deviation shows the most interesting evolution: on the one hand, negative RPMA values correlate with large values of the standard deviation; on the other, when the RPMA is positive an overall linear trend between the standard deviation and the network size is observed, both before and after 2014. Two diversions from the linear trend are observed in correspondence of the dates for which $N \simeq 5 \times 10^4$ and $N \simeq 3 \times 10^5$ and signalling a price decreasing - in fact, the RPMA is negative, here - in two different periods: a first one, started in 2011 and lasted until 2012 and a second one, started in 2014 and lasted until 2016. Then, during the biennium 2016-2017, both the price and the number of nodes are again characterized by an increasing trend; the standard deviation, instead, is smaller and, overall, linearly related to the number of nodes.

Such an evolution is less evident when considering the higher moments of both the in- and the out-degree distributions.

A possible explanation of the phenomenon may be provided by the following mechanism. The Bitcoin ecosystem is highly speculative, hence characterised by rapidly growing and bursting financial bubbles: in this scenario, a negative RPMA roughly identifies the price drawdown following the peak of a price bubble. We hypothesize that a price racing to the top may induce users to adopt a similar behavior, in turn inducing similar connectivity patterns and less disperse degree distributions. On the other hand, during a regime of price decrease, individual choices are influenced by the behavior of the majority to a lesser extent: hence, more heterogeneous connectivity patterns, characterized by broader degree distributions, are more likely to be observed. We refer to this phenomenon as to a network indicator of *herding behavior*: during times of rising bubbles, users tend to hoard money by adopting similar connection patterns - just like an herd of sheep imitating each other behavior[2]; during price drawdowns, instead, the herd is dispersed by financial losses, thus letting the individual behavior emerge.

As the analysis of causality via the Granger method will confirm, this is indeed the case.

3.4 From correlation to Granger-causation

Here, we examine if and, in case, to which extent, the variables we described in the previous section actually 'affect' each other over time: to this aim, we implement a Granger test (see Granger (1969)) to detect the presence of causal relationships among them - a methodology borrowed from econometric (see Gradojevic (2014) and Hong et al. (2009)). Granger causality tests are performed in two different fashions: in *mean* and in *tail*. The differences are going to be explained in the following sub-sections.

[2]Naturally, users do not actually observe each other; however, there are outer incentives to behave similarly.

3.4.1 Granger causality in mean

The simplest Granger test is a *bivariate, causality test in mean*. The test verifies whether a linear model including information about a variable X has a residual sum of squares (hereby, RSS) which is significantly different from the RSS of a model not including the information about X. In formulas, let us consider two stochastic processes X_t and Y_t: the two models - with and without X - are defined as

$$Y_t = \sum_{k=1}^{\tau} \alpha_k Y_{t-k} + \epsilon_t^Y \tag{3.7}$$

and

$$Y_t = \sum_{k=1}^{\tau} \alpha_k Y_{t-k} + \sum_{k=1}^{\tau} \beta_k X_{t-k} + \epsilon_t^{XY} \tag{3.8}$$

where τ is the maximum lag considered and $\{\epsilon_t\}$ is a series of i.i.d. standardised Gaussian random variables. An F-test, then, checks if the variances of ϵ^{XY} and ϵ^X are significantly different: if they are, the null hypothesis of no-causality can be rejected.

3.4.2 Multivariate Granger causality in mean

The multivariate Granger test aims at filtering out the effects of indirect causality as well as bringing to light causal relations that are hidden in the multivariate structure of the data. To understand how it is defined, let us consider a collection of N time series whose length is t $\{X_i(t)\}_{i=1\dots N}$, compactly represented via the tensor $\mathbf{A} \in \mathbb{R}^{N\times t}$; they define the following VAR model

$$\mathbf{A}_t = \sum_{k=1}^{\tau} \mathbf{B}_k \mathbf{A}_{t-k} + \boldsymbol{c} + \boldsymbol{\Xi}_t \tag{3.9}$$

where $\mathbf{B}_k \in \mathbb{R}^{N\times N}$ is a matrix of coefficients, $\mathbf{c}$ is a vector of constants and $\{\boldsymbol{\Xi}_t\}$ is a vector of multivariate standardised Gaussian random variables, at step t. In this framework, we say that the j-th time series X_j is

Granger caused by the i-th time series X_i if at least one time snapshot k exists in correspondence of which $\mathbf{B}_k(j,i)$ is significantly different from 0. For this analysis we have considered $\tau = 4$ for the weekly networks and $\tau = 7$ for the daily ones, i.e. a month-lasting and a week-lasting time window, respectively.

The results of the multivariate Granger test are showed in figure 9. The selected network variables are the following: the number of nodes N, the number of links L, the standard deviation σ, the skewness γ and the kurtosis κ of in- and out-degrees. The financial variable we considered, instead, is the log-return of the Bitcoin price in USD, i.e.

$$r_t = \log_{10}\left(\frac{p_t}{p_{t-1}}\right). \tag{3.10}$$

Importantly, all considered network representations (i.e. BANs and BUNs at both the daily and weekly time scale) have been split in two different sub-samples over which the test has been performed, i.e. before and after 2014 to account for the Mt. Gox failure.

For what concerns the period 2010-2013, all higher moments of the out-degree distribution provide information about future price movements - although each one for a different representation and at a different time scale: more precisely, the standard deviation of out-degrees has a 'negative' effect on price, meaning that an increasingly heterogeneous distribution of the number of payments causes a price drop (and vice versa). On the reverted side, the price influences all moments of the total degree distribution, across different representations and time scales, as well as the number of nodes and the number of links.

Of particular interest is the relationship between the price log-return and the number of nodes. In fact, the presence of a positive feedback loop, at the weekly time scale, can be observed, pointing out that a price increase causes the number of nodes to increase as well and vice versa. At the daily time scale, instead, while a price increase causes an increase of the number of nodes, an increase of the number of nodes causes a price decrease. Hence, as a byproduct, our analysis reveals the presence

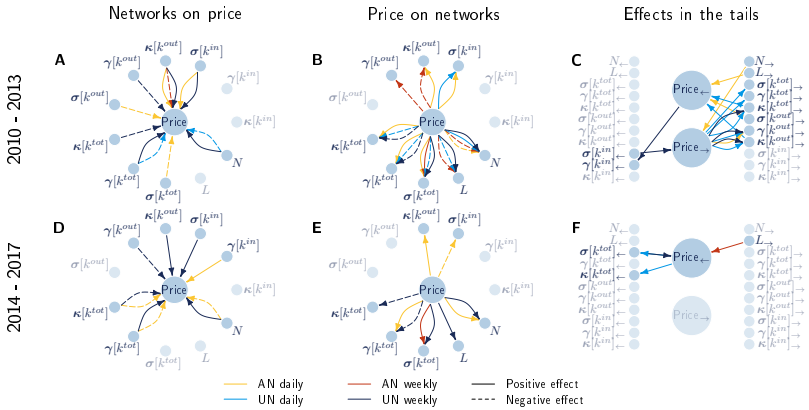

Figure 9: Analysis of the conditional Granger causality structure in the data. Top and bottom panels illustrate the causal relations for the periods 2010-2013 and 2014-2017, respectively. The left, centre and right columns respectively show the effects of the network properties on price, the effects of price on the network properties and the restricted analysis to the tail values, respectively. While all the higher moments of the out-degree distribution provide information on future price movements, price plays a major role in anticipating the moments of the distribution of total degrees, for all representations at all time scales. Figure from Bovet et al. (2019).

of both a slow and a fast dynamics, implying that the Bitcoin ecosystem behaves differently at different time scales.

A second loop involves the kurtosis of out-degrees: on both subsamples, an increase of the out-degree kurtosis implies an increase of the price return; upon considering that an increase of the price return positively affects the number of nodes, the positive feedback loop involving the price return, the number of nodes and the out-degree kurtosis is 'closed'[3].

By comparing the market structure in the period 2010-2013 with the market structure in the period 2014-2017, we observed that the causality structure is remarkably consistent for the BUNs at the weekly time scale

[3]Consistently, we know from figure 8 that the number of nodes and the out-degree kurtosis are positively correlated.

but changes significantly at the daily time scale. More specifically, no causality relationships are detectable, for the daily BUNs, in the period 2014-2017: this result can be interpreted as a signature of increased market efficiency that, in turn, leads to price unpredictability; conversely, structural quantities computed for the BANs still provide information about price movements at the daily time scale.

3.4.3 Granger causality in tail

Beside testing causality *in mean*, one can also test causality *in tail*, with the aim of studying the causality of extremal events, i.e. the ones belonging to either the 10% (left tail) or the 90% (right tail) empirical conditional quantile (see Hong et al. (2009) and Bovet et al. (2019)).

As figure 9 shows, the effects 'in tail', mainly observed during the period 2010-2013, happen at the daily time scale and lead to identify the cause of extreme price movements in the left tail, i.e. what can anticipate a sudden crash in the market: more specifically, as the standard deviation of out-degrees increases, the Bitcoin price decreases, i.e. as the out-degree distribution becomes more heterogeneous (at the daily time scale), a sudden price drop is likely to follow.

The analysis of extremal events also reveal the following relationships: 1) an (extreme) increase of the price return causes an (extreme) increase of the kurtosis of the out-degrees (at both the daily and the weekly time scales) and 2) an (extreme) increase of the kurtosis causes an (extreme) decrease of the price (at the daily time scale). Once combined with the result concerning the feedback loop between the price and the kurtosis, revealed by the analysis of the Granger causality in mean, the second relationship seems to suggest that the loop between the price and the kurtosis may reach a critical limit in correspondence of which a sudden price fall is observed, thus triggering the appearance of a financial crisis.

3.5 Temporal z-scores

To gain further insight into the relationship between the heterogeneity of the out-degree distributions and the Bitcoin price, let us compute the temporal z-score of the standard deviation of the out-degrees. The temporal z-score of a quantity X is defined as

$$z_t[X] = \frac{X - m_t[X]}{s_t[X]} \tag{3.11}$$

where $m_t[X]$ and $s_t[X]$ stand for the sample average and the sample standard deviation of X over the sub-series of data between (the present) time t and a fixed number of steps τ before: in the present analysis, $X \equiv \sigma_t[k^{out}]$ and the rolling window is of one year. The interpretation of the z-score requires the assumption that $\sigma_t[k^{out}]$ is normally distributed: if this is the case, values of the z-score within ± 1, ± 2 and ± 3 occur with a probability amounting at $\simeq 68\%$, $\simeq 95\%$ and $\simeq 99\%$ respectively; moreover, largely positive/negative values of z_t indicate outlier empirical observations, suggesting an ongoing structural change.

Figure 10 shows the temporal z-score of the out-degrees standard deviation for the BUNs, at the weekly time scale - as usual, points are coloured according to the RPMA values. During the period 2010-2013, as the RPMA rises, larger-than-expected values of the z-score are observed, thus implying that values of the out-degrees standard deviation, larger than the arithmetic mean computed over the preceding year, can be observed[4]; on the other hand, the trend of the z-score reverts in correspondence of price drawdowns.

During the period 2015-2016, instead, shifts are observed that are not clearly correlated with price movements: in fact, from 2014 to 2016 a unique price drawdown, de-synced from the temporal z-score, can be observed; this, in turn, suggests that the system is undergoing some kind

[4]Notice that a large value of z_t doesn't necessarily imply a large absolute value of the variable under analysis: it only indicates that the value of the variable under analysis is increasing with respect to the points constituting the sample over which the temporal average is computed.

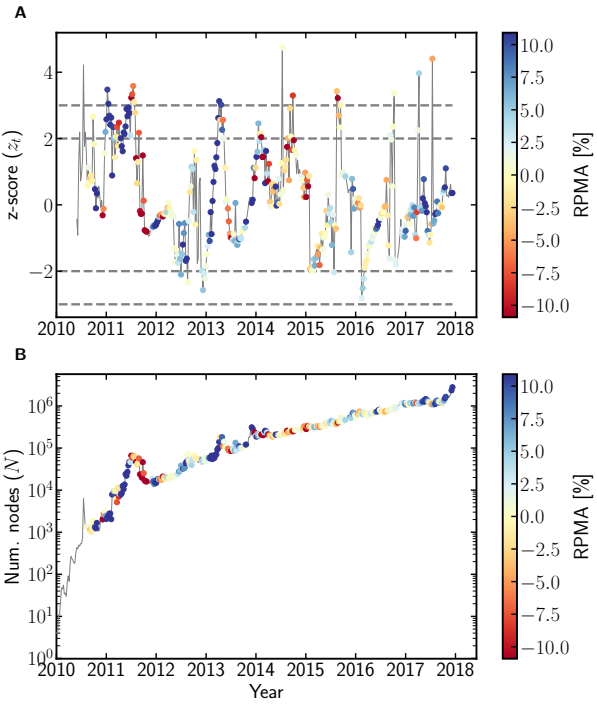

Figure 10: Evolution of the z-score of the out-degrees standard deviation and of the number of nodes for the BUNs, at the weekly time scale. During the period 2010-2013, the z-score of the out-degrees standard deviation grows 'together' with price; drawdowns, instead, appear as periods during which the $\sigma[k^{out}]$ decreases. Moreover, our results reveal peaks between 2015 and 2016, evidencing ongoing structural changes missed by purely financial indicators as the RPMA. Interestingly, since 2017, a price surge is (again) matched by an increase of the temporal z-score of the out-degrees standard deviation. Figure from Bovet et al. (2019).

of structural change, not noticeable by just looking at the price.

Finally, a long period of price growth has started since 2017: even if the values of the temporal z-score remain within the 'non-significance'

interval $[-1, +1]$, we notice positive outliers that point out the presence of snapshots in correspondence of which the out-degrees standard deviation is largely significant.

Figure 10 also shows the evolution of the total number of nodes in a different, yet useful, fashion with respect to figure 7. It confirms our previous comments on the presence of time intervals during which our networks behave differently. In particular, during the periods 2010-2013 and 2017-2018, a rise of the number of nodes is matched by a price surge and viceversa; during the intermediate period 2014-2016, instead, this correlation seems to be less evident. Analogously, for what concerns the trend of higher-order moments of the degree distributions.

Chapter 4

Bitcoin at the mesoscale

The content of this chapter partly overlaps with that of the paper authored by Vallarano et al. (2020). It focuses on the description of the Bitcoin Transaction Networks at the mesoscale, investigating the correlations between degrees, the presence of weakly- and/or strongly-connected components, of a bow-tie structural organization as well as the statistical significance of the latter (seen as a special case of a 'core-periphery' network). For what concerns this specific part of the analysis, we have focused on the BUNs at the weekly time scale.

4.1 Assortativity

The simplest, yet most informative, non-local quantity is represented by the *assortativity*, a measure quantifying the correlations between nodes. A network is said to be *assortative* when the degree correlations are *positive*, i.e. nodes tends to connect to vertices with similar degree (loosely speaking, 'hubs with hubs and leaves with leaves'); on the other hand, a network is said to be *disassortative* when the degree correlations are *negative*, i.e. nodes tends to connect to vertices with different degree (loosely speaking, 'hubs with leaves', as in a star-like configuration).

Let us now consider a bunch of quantities that have been proposed, so far, to measure assortativity. The first one is represented by the standard Pearson correlation coefficient r, computed on the 'excess degrees'.

Following Newman (2003), when undirected networks are considered, one can define the coefficient

$$r_{und} = \frac{\sum_{j,k} jk(e_{jk} - q_j q_k)}{\sigma_q^2} \tag{4.1}$$

where the sums run over the 'excess degrees' (intuitively, imagine to reach a vertex by following a specific edge: the 'excess degree' of the vertex equals 'the vertex degree minus one', i.e. $k-1$), q_k is the 'excess degree' probability distribution, reading

$$q_k = \propto p_{k+1} \tag{4.2}$$

(with p_{k+1} being the plain degree distribution), σ_q^2 is its standard deviation[1] and e_{jk} is the fraction of edges in the network connecting nodes of degree j with nodes of degree k. Naturally, $\sum_j e_{jk} = q_k$.

When considering directed networks, instead, four variants of the aforementioned Pearson coefficient can be calculated, i.e. the ones accounting for the correlation between out-degrees and out-degrees, out-degrees and in-degrees, in-degrees and out-degrees, in-degrees and in-degrees. For example, one of thee variants reads

$$r_{dir}^{out-in} = \frac{\sum_{j,k} jk(e_{jk} - q_j^{out} q_k^{in})}{\sigma_{q^{out}} \sigma_{q^{in}}} \tag{4.3}$$

where e_{jk} now represents the percentage of edges starting from nodes whose out-degree is j and ending on nodes whose in-degree is k. Naturally, it also holds true that $\sum_j e_{jk} = q_k^{in}$.

Plotting the evolution of the aforementioned coefficients on our BUNs shows their weakly disassortative nature (see figure 11). In particular, since r_{dir}^{out-in} is 'asymptotically' zero, one can conclude that $e_{jk} \simeq q_j^{out} q_k^{in}$ - and analogously for the other indices of direct assortativity (the years until 2011 can be considered as a 'transient' period where the Bitcoin ecosystem was still of reduced dimensions, hence sensitive to even small structural changes).

[1] It is there to ensure that $r_{und} \in [-1, 1]$.

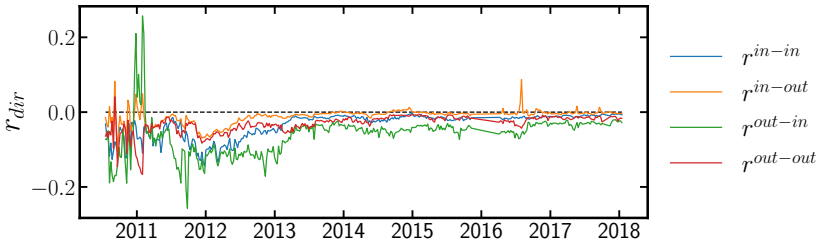

Figure 11: Evolution of the four directed variants of Newman's assortativity coefficient, revealing the (weakly) disassortative character of our BUNs. Moreover, since r_{dir}^{out-in} is 'asymptotically' zero, one can conclude that $e_{jk} \simeq q_j^{out} q_k^{in}$ (and analogously for the other indices).

If, as suggested by figure 11, the fraction of directed links e_{jk} - from any of the nodes whose out-degree is equal to j to any of the nodes whose in-degree is equal to k - is assumed to be well described by the product $q_j^{out} q_k^{in}$, one finds that

$$\begin{aligned} e_{jk} &= \frac{E_{jk}}{L} \propto \left(\frac{j \cdot n_j}{N\bar{j}}\right)\left(\frac{k \cdot n_k}{N\bar{k}}\right) = \left(\frac{j \cdot n_j}{N \cdot L/N}\right)\left(\frac{k \cdot n_k}{N \cdot L/N}\right) = \\ &= \left(\frac{j \cdot n_j}{L}\right)\left(\frac{k \cdot n_k}{L}\right) \Longrightarrow \frac{E_{jk}}{n_j n_k} \propto \frac{j \cdot k}{L} \end{aligned} \tag{4.4}$$

i.e. that the ratio between the number of directed links from nodes whose out-degree equals j to nodes whose in-degree equals k (i.e. E_{jk}) and the total number of such pairs (given by the product $n_j n_k$ between the number of nodes, n_j, whose out-degree is j and the number of nodes, n_k, whose in-degree is k) is proportional to the product of the two degrees. Notice that, in a generic probabilistic framework, this result can be recovered upon calculating

$$\langle E_{jk} \rangle = \sum_{l|k_l^{out}=j} \sum_{m|k_m^{in}=k} p_{lm} \tag{4.5}$$

with $p_{lm} = \frac{k_l^{out} k_m^{in}}{L}$: if this is the case[2], in fact, $\langle E_{jk} \rangle = n_j n_k \frac{j \cdot k}{L}$.

[2]This model is known as Chung-Lu model - see also chapter 5.

An alternative measure of assortativity is provided by the Average Nearest Neighbours Degree (ANND), which is nothing else than the arithmetic mean of the degrees of the neighbors of a node. For undirected networks it reads

$$k_i^{nn} = \frac{\sum_{j(\neq i)=1}^{N} a_{ij} k_j}{k_i}, \quad \forall\, i; \tag{4.6}$$

when directed networks are considered, instead, four variants can be defined, i.e.

$$k_i^{out,out} = \frac{\sum_{j(\neq i)=1}^{N} a_{ij} k_j^{out}}{k_i^{out}}, \quad \forall\, i \tag{4.7}$$

$$k_i^{out,in} = \frac{\sum_{j(\neq i)=1}^{N} a_{ij} k_j^{in}}{k_i^{out}}, \quad \forall\, i \tag{4.8}$$

$$k_i^{in,out} = \frac{\sum_{j(\neq i)=1}^{N} a_{ji} k_j^{out}}{k_i^{in}}, \quad \forall\, i \tag{4.9}$$

$$k_i^{in,in} = \frac{\sum_{j(\neq i)=1}^{N} a_{ji} k_j^{in}}{k_i^{in}}, \quad \forall\, i \tag{4.10}$$

with a clear meaning of the symbols. Scattering the ANND values versus either the in- or out-degrees provides an indication about the network (dis)assortativity: whereas an increasing trend would signal the presence of an assortative behaviour, a decreasing one would, instead, signal the presence of a disassortative behaviour. Figure 12 shows the trend of $k_i^{out,out}$ and $k_i^{in,in}$ for four different snapshots, chosen to depict our BUNs before and after two large price drawdrowns, which took place in 2012 and 2013, respectively. Overall, the disassortative character of our BUNs is evident and can be explained by recalling that hubs are present, i.e. very connected nodes representing exchange markets (or similar institutions) which 'attract' the majority of users (and of their transactions): as an example, let us consider the trend of $k_i^{out,out}$, indicating that nodes with many outgoing connections (e.g. huge market

whales) point to nodes with few outgoing connections (e.g. everyday users) - and similarly for the other indices.

In order to say something about the statistical significance of our findings, let us compare the observed trends of our ANNDs with the expected ones, computed under a suitably defined null model. To this aim we have considered the family of maximum-entropy ones, whose members provide recipes for randomizing a network while preserving part of its structure.

The simplest model is the *Directed Binary Random Graph Model* (DBRGM), according to which

$$p_{ij} = \frac{L}{N(N-1)} \equiv p \tag{4.11}$$

that coincides with the link density; hence, $p = d = O(N^{-1})$ and

$$\langle k_i^{out,out} \rangle_{\text{DBRGM}} = \frac{\sum_{j(\neq i)=1}^{N} p_{ij} \langle k_j^{out} \rangle_{\text{DBRGM}}}{\langle k_j^{out} \rangle_{\text{DBRGM}}} = \frac{L}{N} = O(1) \tag{4.12}$$

since $\langle k_i^{out} \rangle_{\text{DBRGM}} = \langle k_i^{in} \rangle_{\text{DBRGM}} = p(N-1) = \frac{L}{N} = O(1)$ and $\langle k_i^{out,out} \rangle_{\text{DBRGM}} = p(N-1) = \frac{L}{N} = \frac{\sum_{i=1}^{N} k_i^{out}}{N} = \frac{\sum_{i=1}^{N} k_i^{in}}{N} = \overline{k} = O(1)$ as well. However, it predicts completely flat trends, hence being not suited for reproducing trends as the ones characterizing Bitcoin.

A similar drawback is encountered by implementing the probabilistic model according to which $p_{ij} = \frac{k_i^{out} k_j^{in}}{L}$: in this case, in fact,

$$\langle k_i^{out,out} \rangle_{\text{CL}} \simeq \frac{\sum_{j(\neq i)=1}^{N} (k_j^{out})^2}{L}, \quad \forall\, i \tag{4.13}$$

i.e. the predicted trend is again flat - in fact, the closer the Newman's coefficient to zero, the flatter the trend predicted by the Chung-Lu model.

Thus, we have implemented the *Directed Binary Configuration Model* (DBCM) that preserves the in- and out-degree sequences of a network while randomizing everything else: the DBCM predicts values for our ANNDs reading

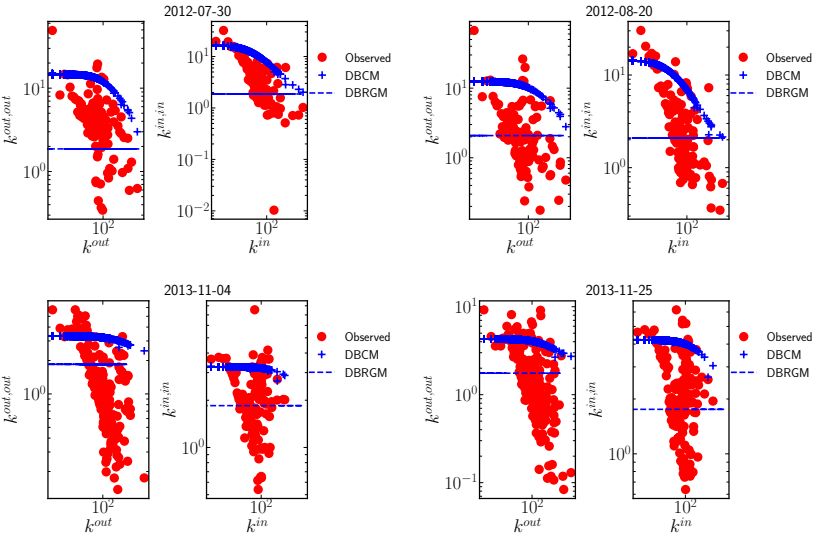

Figure 12: Scattering $k_i^{out,out}$ and $k_i^{in,in}$ versus the out- and the in-degrees, respectively, provides an indication about the network (dis)assortativity: a decreasing trend signals the presence of a disassortative behaviour. The trends predicted by the DBRGM are, with no surprise, flat; however, also the ones output by the DBCM fail to reproduce the empirical clouds of points, predicting a network that is less disassortative than observed. As solely enforcing the degree sequences is not enough to reproduce the degree correlations of our BUNs, the observed disassortativity can be interpreted as a genuine signal of the system self-organization.

$$\langle k_i^{out,out} \rangle_{\text{DBCM}} = \frac{\sum_{j(\neq i)=1}^{N} p_{ij} k_j^{out}}{k_i^{out}}, \quad \forall\, i \tag{4.14}$$

$$\langle k_i^{out,in} \rangle_{\text{DBCM}} = \frac{\sum_{j(\neq i)=1}^{N} p_{ij} k_j^{in}}{k_i^{out}}, \quad \forall\, i \tag{4.15}$$

$$\langle k_i^{in,out} \rangle_{\text{DBCM}} = \frac{\sum_{j(\neq i)=1}^{N} p_{ji} k_j^{out}}{k_i^{in}}, \quad \forall\, i \tag{4.16}$$

$$\langle k_i^{in,in} \rangle_{\text{DBCM}} = \frac{\sum_{j(\neq i)=1}^{N} p_{ji} k_j^{in}}{k_i^{in}}, \quad \forall\, i \tag{4.17}$$

where the probability coefficients $\{p_{ij}\}_{i,j=1}^{N}$ have been numerically determined by solving the likelihood equations

$$k_i^{out} = \langle k_i^{out} \rangle_{\text{DBCM}} = \sum_{j(\neq i)=1}^{N} p_{ij} = \sum_{j(\neq i)=1}^{N} \frac{x_i y_j}{1 + x_i y_j}, \quad \forall\, i \quad (4.18)$$

$$k_i^{in} = \langle k_i^{in} \rangle_{\text{DBCM}} = \sum_{j(\neq i)=1}^{N} p_{ji} = \sum_{j(\neq i)=1}^{N} \frac{x_j y_i}{1 + x_j y_i}, \quad \forall\, i \quad (4.19)$$

(and the degrees are reproduced by definition). For more details on the definition of the entropy-based null models, see chapter 5.

The expected trends of $k_i^{out,out}$ and of $k_i^{in,in}$, obtained via the aforementioned procedure, are shown in blue, in figure 12: notice how they fail in capturing the observed trends, predicting networks that are less disassortative than the empirical ones. In other words, enforcing the degree sequences is not enough to reproduce the degree correlations of our BUNs; on the contrary, the DBCM would predict a configuration that is more homogeneous than the observed one. Henceforth, Bitcoin disassortativity - at the level of the weekly BUNs - can be interpreted as a genuine signal of the system self-organization.

4.2 Connected components

Before looking at Bitcoin from a proper mesoscale perspective, let us give some definitions that will be useful in the following (see Latora et al. (2017)):

Definition 1 (Walks)**.** *Given a graph $G(V, E)$, a* walk *$W(x, y)$ from node x to node y is an alternating sequence of nodes and edges that begins at x and ends at y, i.e. $W = (n_0 = x, e_1, v_1 \dots e_k, v_k = y)$ such that $e_j = (v_{i-1}, v_i) \in E$.*

Definition 2 (Paths)**.** *A* path *is a walk in which no node is visited more than once.*

Definition 3 (Strong connectedness)**.** *Two nodes x and y in a directed graph G are said to be* strongly connected *if there exist a path from x to y and a*

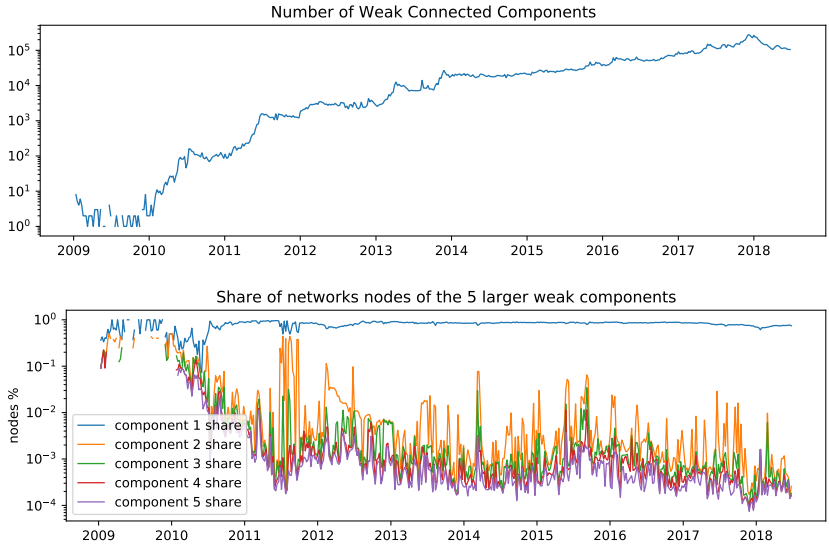

Figure 13: Evolution of the number of weakly connected components (top panel) and of the size of the top five WCCs, calculated as a percentage of the total number of nodes N (bottom panel).

path from y to x. A directed graph is said to be strongly connected *if all pairs of nodes are strongly connected. A* strongly connected component *of G associated with node x is the maximal strongly connected induced subgraph containing node x.*

Definition 4 (Weak connectedness)**.** *The undirected graph G_u obtained by removing all directions from the arcs of G is called the* underlying, undirected graph *of G. A directed graph G is said to be* weakly connected *if G_u is connected. A* weakly connected component *of G is a component of its underlying, undirected graph G^u.*

Let us now inspect the evolution of the number and of the size of the weakly connected components (WCCs) characterizing our BUNs. As figure 13 shows, although a large number of them is visible throughout the entire Bitcoin history (there are more than 10^5 different WCCs in 2018), a giant WCC emerges after 2012 - in fact, no other connected component is comparable, in size, with the largest one after this date.

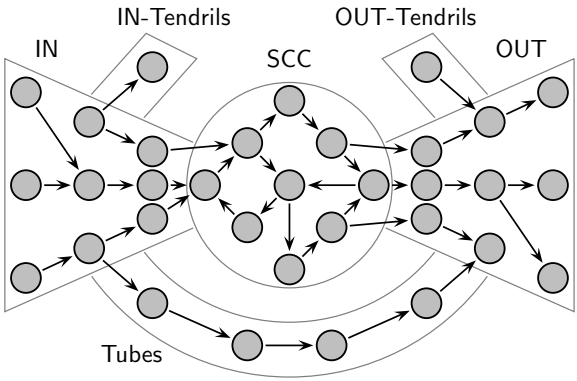

Figure 14: Pictorial representation of a bow-tie structure. Figure from Glatterfelder (2019).

The large number of small WCCs may be a consequence of the misclassification of addresses, that have not been (correctly) assigned to the same user(s): hence, there may be one-time transactions among individuals, transfers of money between different wallets controlled by the same person, etc. that give origin to isolated sets of nodes. However, transactions like these represent an overall small percentage of their total number: in fact, the vast majority of transactions contribute to shape the largest WCC throughout the entire Bitcoin history. As we already observed, the Bitcoin economy is very interconnected, thanks to the presence of large hubs 'through' which most of the transactions pass.

The presence and the evolution of the size of a giant strongly connected component (SCCs) is, instead, related to the presence of the so-called *bow-tie* structure: for this reason, we study it in a dedicated section (the next one).

4.3 Bow-tie structure

The definition of *bow-tieness* rests upon the concept of *reachability*: we say that j is *reachable* from i if a path from i to j exists[3]. Mutual reachability represents an equivalence relation on the vertices of a graph, the equivalence classes being the strongly connected components of the graph itself. Hence, the bow-tie decomposition of a graph consists of the following sets of nodes (see Lidth de Jeude et al. (2019)):

- SCC $\equiv S$; it is the SCC. Each node within the SCC can be reached by any other node within it. This means that a directed path exists connecting each node with any other node;
- IN $\equiv \{i \in V \setminus S \mid i \to S\}$; each node within IN can reach S;
- OUT $\equiv \{i \in V \setminus S \mid S \to i\}$; each node within OUT can be reached by S;
- TUBES $\equiv \{i \in V \setminus S \cup \text{IN} \cup \text{OUT} \mid \text{IN} \to i \text{ and } i \to \text{OUT}\}$; each node within TUBES can be reached by IN and can reach OUT;
- IN-TENDRILS $\equiv \{i \in V \setminus S \mid \text{IN} \to i \text{ and } i \nrightarrow \text{OUT}\}$; each node within IN-TENDRILS can be reached by IN but cannot reach OUT;
- OUT-TENDRILS $\equiv \{i \in V \setminus S \mid \text{IN} \nrightarrow i \text{ and } i \to \text{OUT}\}$; each node within OUT-TENDRILS cannot be reached by IN but can reach OUT;
- OTHERS $\equiv \{i \in V \setminus S \cup \text{IN} \cup \text{OUT} \cup \text{TUBES} \cup \text{IN-TENDRILS} \cup \text{OUT-TENDRILS}\}$.

Figure 14 provides a pictorial representation of a bow-tie structure and figure 15 shows the evolution of the size of its components. Generally speaking, a large SCC, incorporating the vast majority of nodes, starts emerging in 2012, 'stabilizes' around mid-2013 and persists until

[3] A directed graph is said to be *strongly connected* if any two nodes are mutually reachable: notice the similarity with the definition provided at the beginning of the previous section.

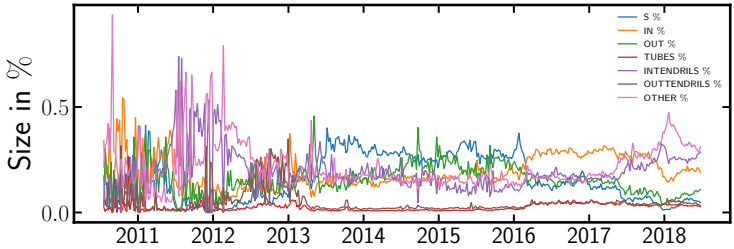

Figure 15: Evolution of the percentages of nodes belonging to the various components of a bow-tie structure. During the biennium 2012-2013, the SCC steadily rises until it reaches $\simeq 30\%$ of the network size; afterwards, it remains quite constant until 2016 when it starts shrinking and the percentage of nodes belonging to it goes back to the pre-2012 values. Moreover, during this last period, both the SCC and the OUT-component shrink, while the IN-component becomes the dominant portion of the network.

2016. More specifically, during the biennium 2012-2013 the SCC steadily rises until it reaches $\simeq 30\%$ of the network size; afterwards, during the biennium 2014-2015, it remains quite constant; then, during the last two years covered by our data set (i.e. 2016-2018), it shrinks and the percentage of nodes belonging to it goes back to the pre-2012 values. While in the biennium 2014-2016 the percentage of nodes constituting the SCC is larger than the percentage of nodes belonging to the other subsets, since 2016 this is no longer true: in fact, while both the SCC and the OUT-component shrink, the IN-component becomes the dominant portion of the network.

Different results have been reported in Di Francesco Maesa et al. (2019): however, this may be due to the different data collection and data mining processes implemented there.

4.4 Core-periphery structure

In the previous section we have described the evolution of the bow-tie structure of our BUNs. Let us now ask ourselves if it induces a statistically significant *core-periphery* structure. The latter one is a kind of

mesoscale structure that partition the node set in two, i.e. into a core (a densely inter-connected set of vertices) and a periphery (the remaining, loosely inter-connected, set of vertices); naturally, core nodes are still connected to the periphery ones (see Rombach et al. (2014)).

In order to detect the presence of a core-periphery structure we ran a recently proposed method (see Jeude et al. (2019)) based on the extension of the *surprise* score function. Originally proposed to detect communities (see Aldecoa et al. (2013a) and Aldecoa et al. (2013b)), surprise reads

$$\mathscr{S} = \sum_{l \geq l^*_\bullet}^{\min\{L, V_\bullet\}} \frac{\binom{V_\bullet}{l}\binom{V_\circ}{L-l}}{\binom{V}{L}} \tag{4.20}$$

where

- V is the total number of node pairs (in case of directed networks, $V = N(N-1)$);
- $V_\bullet$ is the number of intra-cluster node pairs (i.e. the number of node pairs inside clusters) while $V_\circ$ is the number of inter-cluster node pairs (i.e. the number of node pairs between clusters);
- $l^*_\bullet$ is the number of observed intra-cluster links (i.e. the number of links inside clusters);
- L is the total number of links in the network.

From a statistical point of view, $\mathscr{S}$ is the p-value of an hypergeometric distribution. The latter describes the probability of observing l intra-cluster links out of the total L ones, i.e. of obtaining l successes out of a total number of L draws, without replacement, from a population where $V_\bullet$ occurrences have the desired feature: in our case, being an intra-cluster node pair. The lower this probability, the more 'surprising' the partition is: if it is found to lie below a given threshold one can conclude that the test rejects the hypothesis that links are distributed randomly[4].

[4]More precisely, according to the Directed Binary Random Graph Model.

While traditional surprise focuses on intra-cluster and inter-cluster links, in order to detect a core-periphery structure, the surprise score must be extended to account for a tripartite division of links: core links, periphery links and links between the core and the periphery. In Jeude et al. (2019) the authors propose to minimize

$$\mathscr{S}_{\parallel} = \sum_{i \geq l_{\bullet}^{*}} \sum_{j \geq l_{\circ}^{*}} \frac{\binom{V_{\bullet}}{i}\binom{V_{\circ}}{j}\binom{V-(V_{\bullet}+V_{\circ})}{L-(i+j)}}{\binom{V}{L}} \tag{4.21}$$

i.e. the p-value of a multivariate hypergeometric distribution, describing the probability of observing $i + j$ successes out of L draws, without replacement, from a population where $V_{\bullet}$ objects are of a first kind (in our case, being core links) and $V_{\circ}$ objects are of a second kind (in our case, being periphery links). It can be proven that minimizing the multivariate surprise is equivalent at defining the partition which is least likely to be explained by the Directed Binary Random Graph Model with respect to the Directed Binary Stochastic Block Model.

The application of the multivariate surprise to the partition induced by the bow-tie structure reveals that it indeed induces a significant core-periphery structure ($\mathscr{S}_{\parallel}$ is steadily below the threshold of 5%), the core being the SCC and the periphery being composed by all the other nodes. Figure 16 shows the evolution of the percentage of nodes composing the core (i.e. the SCC of the bow-tie) and the periphery (i.e. all the remaining portions) of our BUNs. As expected from the results concerning the SCC, the periphery contains the vast majority of nodes throughout the entire Bitcoin history.

While inspecting the relationship between purely structural quantities and purely financial indicators as the Bitcoin price can be done by explicitly showing the RPMA values per snapshot, another approach is that of partitioning the Bitcoin history in time windows characterized by the rate of growth of the BTC price, i.e. the bubbles introduced in chapter 1. As figure 16 shows, the size of the core (and that of the periphery as well) 'reacts' to the transition between a period of price growth and a period of price decrease. This is rather clear for the first two bubbles where

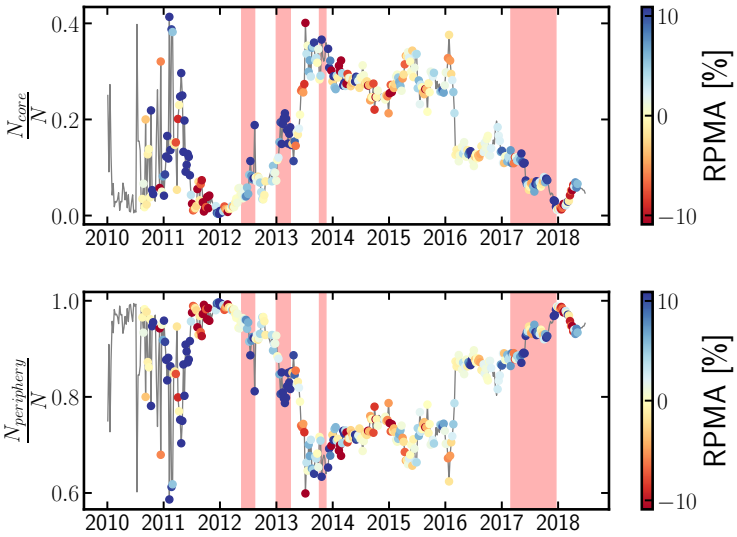

Figure 16: Evolution of the percentage of nodes composing the core and the periphery of our BUNs. Each dot is coloured according to the value of the RPMA at that time. Shaded areas indicate periods during which the price grows.

positive a increase of the BTC price coincides with an enlargement of the core; during the last, and longer, bubble (of which we have highlighted only the last six months), the behaviour seems to be somehow inverted: a decrease of the core size coincides with a price growth of the price, while the trend reverts just after the bubble crashes (i.e. the price decreases).

Let us now repeat the temporal analysis carried out via the z-scores by considering X as the number of nodes belonging to the core - or, in a complementary fashion, to the periphery. The results are shown in figure 17: from the plots, it is readily seen that values of the z-score larger than $+3$ (either considering the percentage of core nodes or the one of periphery nodes) coincide with price peaks (and are followed by a price decrease). While this is clearly true for the first three bubbles, during the

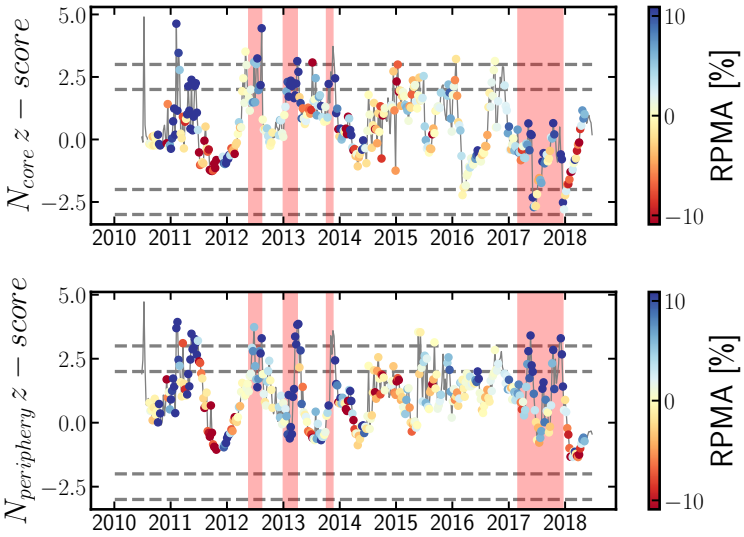

Figure 17: Evolution of the temporal z-score for the number of nodes composing the core (top panel) and the periphery (bottom panel) of our BUNs: the rolling window is of one week. Points are coloured according to the value of the log-return of the Bitcoin price in USD, in that week. Shaded areas indicate periods during which the price grows.

last one, instead, the situation changes: the z-score, in facts, exceeds the -3 threshold several times - two during the actual last six months; three considering the bubble in its entirety.

What emerges from our analysis is that our BUNs are characterized by a core-periphery structure, a deeper analysis of which reveals a certain degree of bow-tieness (i.e. the presence of an SCC, an IN- and an OUT-component and some tendrils attached to the IN-component). Interestingly, the evolution of the BUN mesoscale structure experiences fluctuations that seem to be correlated with the presence of bubbles, i.e. periods of price surge and decline observed throughout the entire Bitcoin history: our results, thus, further confirm the interplay between structural quantities and price movements.

4.5 Centrality and centralization

As we said at the beginning of the present work, one of the goals Bitcoin aimed at achieving was that of *decentralization*: many of the early adopters were, in fact, moved by the idea of getting rid of a central authority authorizing the transactions being done. Decentralization is also related to the concept of *equality*: a decentralised infrastructure is, in principle, one where everybody plays the same role. Although the hardware Bitcoin is built upon (i.e. the blockchain) indeed implements decentralization, it is much less clear if decentralization is recovered at a topological level as well.

Following Lin et al. (2020), we investigate Bitcoin (de)centralization and (in)equality by considering the Gini coefficient of our degree distributions. The Gini coefficient attempts to measure the unevenness of a distribution of a certain quantity[5]: given a set of values $\{c_i\}_{i=1}^N$, the Gini index is defined as

$$G_c = \frac{\sum_{i=1}^N \sum_{j=1}^N |c_i - c_j|}{2N \sum_{i=1}^N c_i} \tag{4.22}$$

and assumes values between 0 and 1; while a Gini index of 0 indicates perfect evenness (e.g. everyone has exactly the same income), a Gini index of 1 indicates perfect unevenness (e.g. a population whose entire income is concentrated in the hands of a single individual). Applying the Gini coefficient to the degrees of our BUNs aims at shedding light on the (un)evenness of the nodes degree centralization: while a value close to 0 would depict an ecosystem where all actors have exactly the same number of interactions with each other, a value close to 1 would indicate that there are nodes participating to the vast majority of transactions.

From the results in figure 18, it appears that, after a period of growth lasted until mid-2013, during which it reached values as large as 0.75, the Gini coefficient has decreased and is now steadily around the value of 0.5. Overall, we would like to stress that 0.5 is not a small value: in fact, it

[5]Usually, it is employed to measure the unevenness of the income distribution.

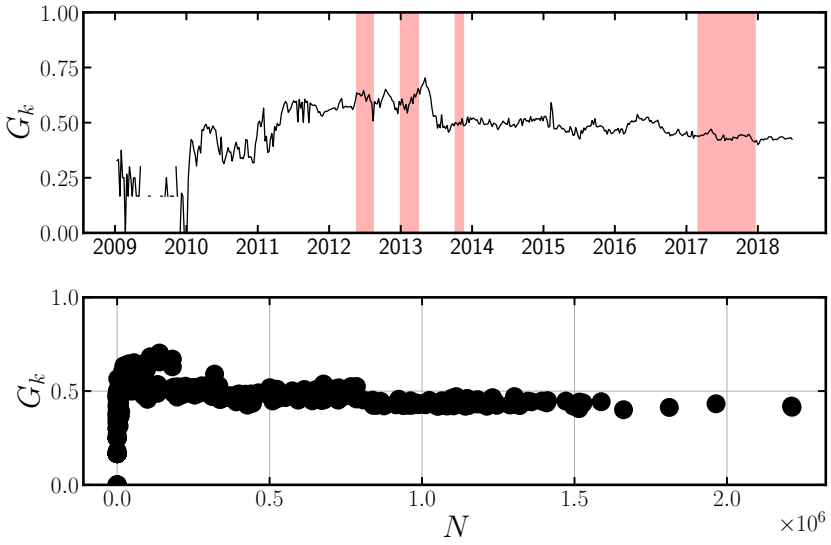

Figure 18: Evolution of the Gini coefficient for the degree distribution of the BUNs plotted versus time (top panel) and versus the total number of nodes (bottom panel). Shaded areas indicate periods during which the price grows. After a period of growth, during which it reached values as large as 0.7, the Gini coefficient has decreased and is now steadily around the value of 0.5, meaning that 50% of connections are incident to the 1% of nodes. Notice also the big leap down in 2013 maybe due to the Mt. Gox 'loss of prominence' in the Bitcoin ecosystem.

describes an ecosystem[6] where the 50% of connections are incident to the 1% of nodes. It is also interesting to notice the big leap down of the Gini coefficient in 2013: during that year, Mt. Gox (which managed $\simeq 70\%$ of transactions at the time Decker et al. (2014)) started the down-ward spiral which eventually led to its bankruptcy in 2014: USD withdrawals halting, financial investigations and expensive lawsuits weakened the trading website ability to stay on the market. The final blow was the public discovery of a huge theft of around 750.000 bitcoins, which went

[6] A 'ring of hubs', where $N_h = 50$ hubs are connected in a ring-like fashion and $N_l = N_h = 50$ leaves are connected to each of them (see also later).

on undetected for years.

The huge decrease of the Gini coefficient may be, thus, related to the Mt. Gox 'loss of prominence' in the Bitcoin ecosystem. On the other hand, bubble periods seem to have little correlation with the evolution of the Gini coefficient.

The evolution of the Gini coefficient may lead us to imagine that the Bitcoin ecosystem has become similar to a very centralized structure, pretty much similar to a star graph, at some point during its history. In order to answer this question, we have computed the so-called *centralization index* at the weekly time scale (from Lin et al. (2020)). Centralization indices are global measures intended to measure the centrality of the entire network (instead of providing a rank of its nodes). In mathematical terms, the centralization reads

$$C_c = \frac{\sum_{i=1}^{N}(c^* - c_i)}{\max\left\{\sum_{i=1}^{N}(c^* - c_i)\right\}} \tag{4.23}$$

where $c^* = \max\{c_i\}_{i=1}^{N}$ represents the empirical, maximum value of the chosen centrality measure (i.e. computed on the network under consideration) and the denominator is calculated over a benchmark graph, defined as the one providing the maximum attainable value of the quantity $\sum_{i=1}^{N}(c^* - c_i)$. Here we consider the degree-centralization index only, the benchmark graph for which is nothing else than a star graph (with the same number of nodes of the network under inspection): hence

$$C_k = \frac{\sum_{i=1}^{N}(k^* - k_i)}{(N-1)(N-2)} \tag{4.24}$$

and the degree centralization would reveal to us if (and, in case, 'how much') Bitcoin has become similar to a star graph at a certain point during its history. In eq. 4.24 k_i is the degree of node i.

While during the initial phases of its life, Bitcoin was indeed quite similar to a star graph, figure 19 reveals that the degree-centralization has quickly stabilized around very small values. Overall, we may, thus, conclude that Bitcoin is not evolving towards a star-like structure, where

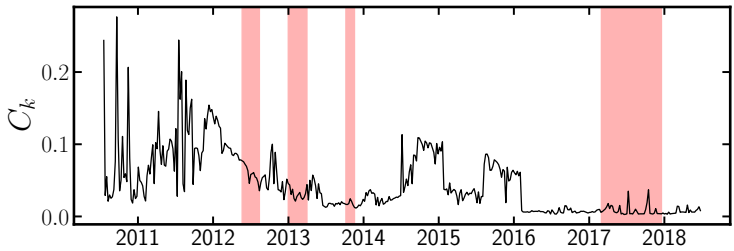

Figure 19: Evolution of the degree-centralization index on our BUNs. Overall, it is quite small, indicating that a star-like configuration indeed oversimplifies the actual topology of our BUNs; on the other hand, a configuration composed by many interconnected stars is compatible with the values shown here. Shaded areas indicate periods during which the price grows.

a single central node participates to all transactions. However, the large value of the Gini coefficient let us suspect that there may be several hubs: hence, the unrealistic picture of a star-like structure may be replaced by the more realistic one depicting several 'locally star-like' structures - similarly to what is observed for the Bitcoin Lightning Network (see chapter 4). The centers of these structures are 'local hubs', i.e. vertices with a large number of connections, that are crossed by a large percentage of paths and that are connected among them.

A toy model can help understanding the two apparently contradictory results provided by the Gini coefficient and the degree-centralization. Imagine N_h hubs connected between them and N_l leaves connected to each of them; hence, the total number of nodes is $N = N_h(N_l + 1)$, the degree of each hub reads $k_h = (N_h - 1) + N_l$, the degree of each leave reads $k_l = 1$ and

$$
\begin{aligned}
G_k &= \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|k_i-k_j|}{2N\sum_{i=1}^{N}k_i} = \frac{2[(N_h-1)+N_l-1]N_h^2N_l}{2N\left[N_h(N_h-1)+2N_hN_l\right]} \\
&= \frac{[(N_h-1)+N_l-1]N_h^2N_l}{N_h(N_l+1)\left[N_h(N_h-1)+2N_hN_l\right]} \\
&= \frac{(N_h+N_l-2)N_l}{(N_l+1)\left[(N_h-1)+2N_l\right]} \simeq \frac{N_h+N_l}{N_h+2N_l};
\end{aligned}
\tag{4.25}
$$

now, $G_k \simeq 2/3$ if $N_l = N_h$, i.e. if each hub is linked to a number of 'leaves' that matches the total number of hubs and $G_k \to 1/2$ as $N_l \to +\infty$, i.e. if the number of leaves per hub becomes 'very large'[7]. In this setting, we have that

$$
C_k = \frac{\sum_{i=1}^{N}(k^*-k_i)}{(N-1)(N-2)} = \frac{[(N_h-1)+N_l-1]N_hN_l}{(N_h(N_l+1)-1)(N_h(N_l+1)-2)} \simeq \frac{N_h+N_l}{N_hN_l}
\tag{4.26}
$$

which amounts at $C_k \simeq 0.02$ if we set $N_h = N_l = 100$. Hence, by opportunely tuning the parameters of our model we can recover core-periphery structures for which a large Gini coefficient co-exists with a small degree-centralization.

4.6 Dyadic motifs and reciprocity

Let us now move to the analysis of *network motifs*, i.e. sub-graphs composed by at least two nodes whose abundance is usually reflected into some functional properties of the system under analysis. Of particular

[7]The 'ring of hubs' represents an interesting alternative: imagine N_h hubs connected in a ring-like fashion and N_l leaves connected to each of them; although the total number of nodes is (still) $N = N_h(N_l+1)$ and the degree of each leave (still) reads $k_l = 1$, the degree of each hub, now, reads $k_h = N_l + 2$; hence, $G_k = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|k_i-k_j|}{2N\sum_{i=1}^{N}k_i} = \frac{(N_h+2-1)N_h^2N_l}{N_h(N_l+1)[N_h(N_l+2)+N_hN_l]} \simeq \frac{N_h}{2N_l}$ and $G_k \simeq 1/2$ if $N_l = N_h$. In this case, half of the connections are incident to a percentage $\frac{N_h/2}{N_h(N_l+1)} \simeq \frac{1}{2N_l}$ of the nodes.

importance are the so-called dyadic motifs: given a generic binary, directed network $G(V, E)$ and any two nodes $i, j \in V$, dyadic motifs are defined as the following, mutually exclusive, occurrences

- a *reciprocated dyad*: when both $(i, j) \in E$ and $(j, i) \in E$. We denote the total number of reciprocated dyads in the network as

$$L^{\leftrightarrow} = \sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} a_{ij} a_{ji}; \tag{4.27}$$

- a *non-reciprocated dyad*: when either $(i, j) \in E$ or $(j, i) \in E$. We denote the total number of non-reciprocated dyads in the network as

$$L^{\rightarrow} = \sum_{i=1}^{N} \sum_{j(>i)=1}^{N} [a_{ij}(1 - a_{ji}) + a_{ji}(1 - a_{ij})]; \tag{4.28}$$

- an *empty dyad*: when both $(i, j) \notin E$ and $(j, i) \notin E$. We denote the total number of empty dyads in the network as

$$L^{\not\leftrightarrow} = \sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} (1 - a_{ij})(1 - a_{ji}). \tag{4.29}$$

The definition of dyads naturally leads to the definition of *reciprocity*, i.e. the ratio between the total number of reciprocated dyads and the total number of links:

$$r = \frac{\sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} a_{ij} a_{ji}}{\sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} a_{ij}} = \frac{L^{\leftrightarrow}}{L}. \tag{4.30}$$

Figure 20 shows the evolution of the reciprocity and of its temporal z-score: overall, the value of r is very low, meaning that our BUNs are not so reciprocated; still, the evolution of the reciprocity shows some peaks of activity in correspondence of the bubbles. This is further confirmed

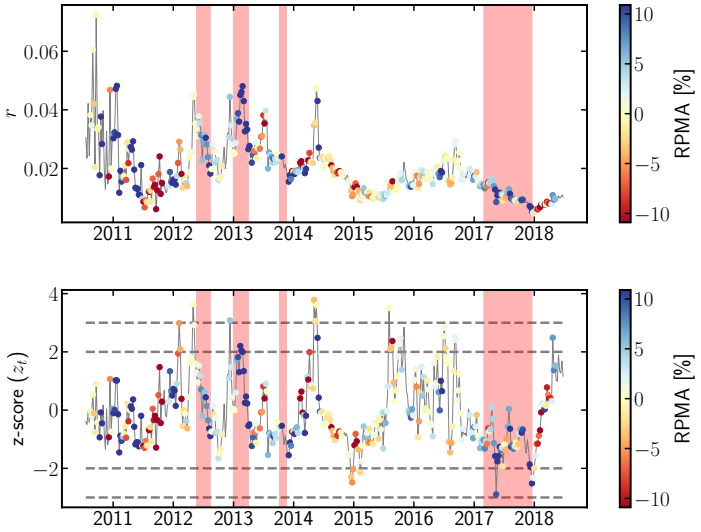

Figure 20: Evolution of the reciprocity (top panel) and of its temporal z-score (bottom panel). The value of r is very low throughout the entire Bitcoin history, meaning that our BUNs are not so reciprocated; the evolution of its temporal z-score, instead, rises significantly in correspondence of the bubbles. Points are coloured according to the value of the log-return of the Bitcoin price in USD, in that week. Shaded areas indicate periods during which the price grows.

by the calculation of the temporal z-score of reciprocity that rises significantly in correspondence of the first three bubbles and decreases during the last one.

Let us now study dyadic motifs by adopting a different approach with respect to the one employed so far. Instead of using the temporal z-score to spot 'temporal' outliers, i.e. values that are statistically significant with respect to a time average, let us consider an index that points out quantities not compatible with a given null model. In order to do so, we will employ the DBCM. In this framework, a z-score of the kind

$$z[X] = \frac{X(\mathbf{A}^*) - \langle X \rangle}{\sigma[X]} \tag{4.31}$$

remains naturally defined, where $X(\mathbf{A}^*)$ is the empirical value of the quantity of interest (i.e. observed on the original network $\mathbf{A}^*$), $\langle X \rangle$ and $\sigma[X]$ are, respectively, its expectation value and its standard deviation, both computed on the ensemble induced by the DBCM. The interpretation of this z-score is the following one: values such that $z[X] > +3$ signal that the empirical value is significantly larger than expected while values such that $z[X] < -3$) signal that the empirical value is significantly smaller than expected. In both cases one may conclude that the empirical value $X(\mathbf{A}^*)$ is not compatible with the specific model and something else is required to fully account for it. On the other hand, if $-3 \leq z[X] \leq +3$, there is no evidence of a significant deviation from the expected value and one may conclude that $X(\mathbf{A}^*)$ is completely explained by the constraints defining the model at hand.

Remarkably, in the case of dyadic motifs, we are able to compute both the expected value and the standard deviation of the number of reciprocal, non-reciprocal and empty dyads analytically (see chapter 5). In figure 23 we can observe the evolution of the empirical values of the dyadic measures, computed on our BUNs, week by week: all measures have a huge spike just before the crash of the last bubble. Overall, it is quite difficult to say how much this is related to the overall growth of the size of our Bitcoin Transaction Networks; while the temporal z-score in figure 20 suggests that r may be affected by some kind of seasonality, it does not tell us anything about its significance in a 'static' fashion - neither about dyads.

To this aim, let us consider the z-score induced by the DBCM, providing information about the extent to which it dyadic values are compatible with the network degree sequences. As figure 23 shows, the z-score for the number of reciprocated dyads is very large, meaning that $L^{\leftrightarrow}$ cannot be explained by just enforcing the degree sequences. The reason is intuitive: given the level of sparseness of our networks, we can expect that observing links, pointing in opposite directions, by chance (i.e. just

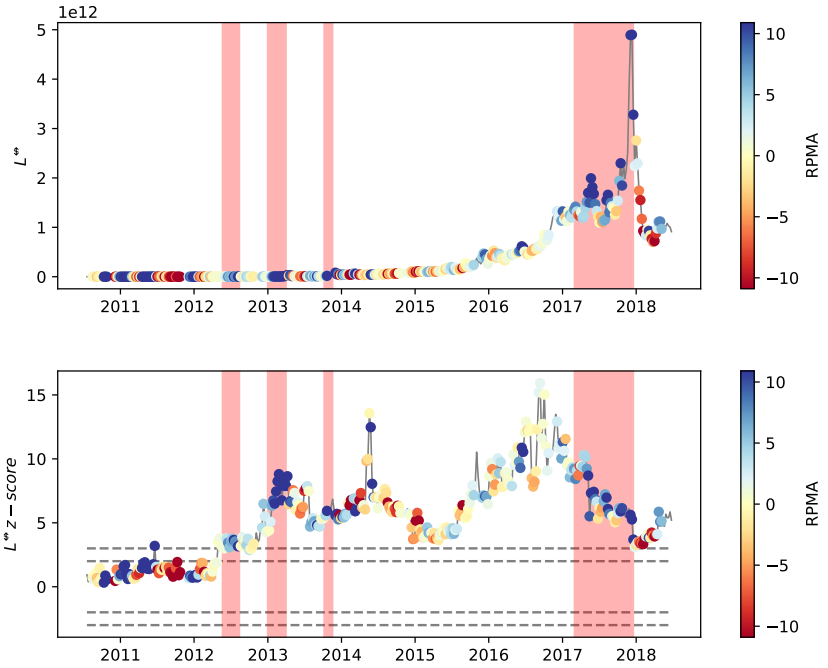

Figure 21: Top figure: absolute number of couples of nodes with no link in between (empty dyads). Bottom figure: evolution of the empty dyads z-score, computed over the ensemble induced by the Directed Binary Configuration Model. The z-score proves that the observed empty dyads are over-represented with respect to the randomized ensemble.

as a consequence of a randomly rewiring the nodes connections) is very unlikely[8]; on the other hand, 'reciprocal' transactions between users[9] can happen quite often, during a week.

The interpretation of the behavior of empty and single dyads is analogous: by chance, a larger-than-observed number of non-reciprocated

[8]To provide a quantitative evaluation of how unlikely it is, consider, that under the DBRGM, the reciprocity reads $\langle r \rangle_{\text{DBRGM}} = p = O(N^{-1})$. Since, for the vast majority of the users, $k_i^{out} \simeq k_i^{in} = O(1)$, $p_{ij}^{\text{DBCM}} = O(N^{-1})$, the 'local' level of reciprocity expected under the DBCM is similar to the one expected under the DBRGM.

[9]Actually, between a user and an exchange market.

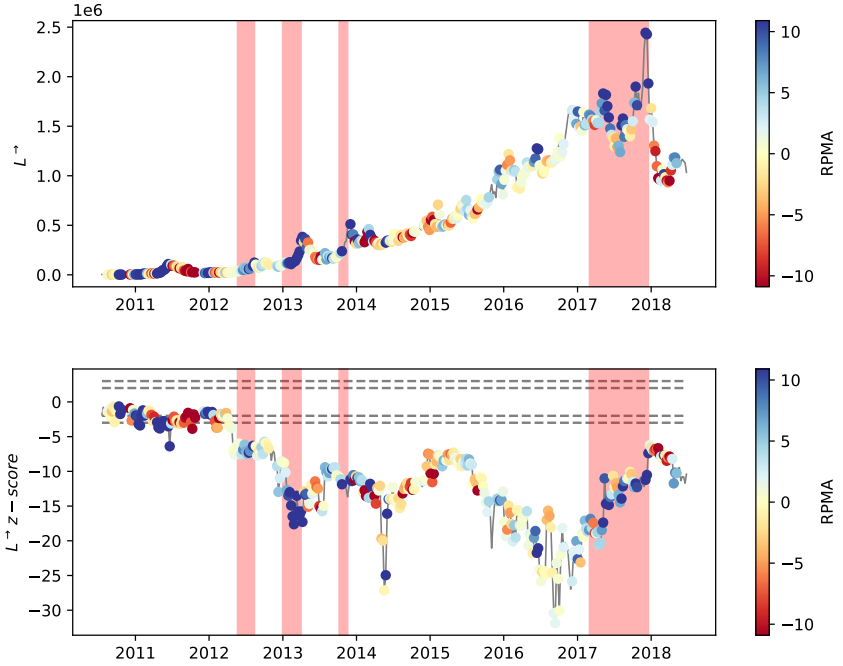

Figure 22: Top figure: absolute number of couples of nodes with exactly one link in between (single dyads). Bottom figure: evolution of the single dyads z-score, computed over the ensemble induced by the Directed Binary Configuration Model.

dyads are created, whence their over-representation within the DBCM ensemble and the negative z-score recovered by our analysis. In order to understand why this implies that the DBCM tends to create less-than-observed empty dyads, let us imagine to 'destroy' a reciprocal dyad, by decoupling the two paired links: in order to create 'more' single dyads, one of the two links mus be redirected towards a previously disconnected node; upon doing so, a reciprocal dyad disappears, as well as an empty dyad, while two single dyads are created.

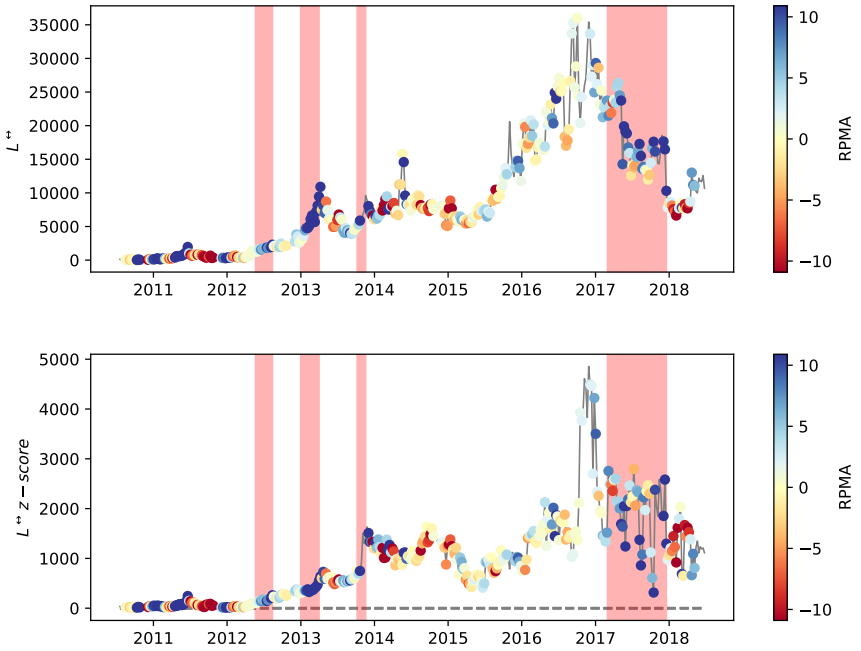

Figure 23: Top figure: absolute number of couples of nodes with reciprocated links in between (full dyads). Bottom figure: evolution of the full dyads z-score, computed over the ensemble induced by the Directed Binary Configuration Model. Given the level of sparseness of our networks, we can expect that observing links, pointing in opposite directions, as a consequence of a randomly rewiring the nodes connections, is very unlikely. Analogously for what concerns the number of empty and single dyads. Intuitively, we can imagine to 'destroy' a reciprocal dyad, by decoupling the two paired links: upon doing so, a reciprocal dyad disappears, as well as an empty dyad, while two single dyads are created. Dashed, gray lines signal the values of ± 2 and ± 3. Points are coloured according to the value of the log-return of the Bitcoin price in USD, in that week. Shaded areas indicate periods during which the price grows.

Chapter 5

The Bitcoin Lightning Network

This chapter is devoted to the description of the Bitcoin Lightning Network (BLN) and partly overlaps with the paper authored by Vallarano et al. (2020). After depicting the basic properties of the BLN, studied as a network of users at the daily time scale, it focuses on its mesoscale structure, revealing it to be significantly centralized.

5.1 The Bitcoin Lightning Network (in brief)

Proposed in 2015 (see Poon et al. (2016)) and launched in 2018, the *Bitcoin Lightning Network* (BLN hereby) is a 'Layer 2' protocol that can operate on top of blockchain-based cryptocurrencies like Bitcoin. It works by creating bilateral channels for *off-chain* payments which are settled concurrently on the blockchain once the channel are closed. The aim is that of allowing any two users to exchange money while requiring neither transaction fees nor any confirmation - thus avoiding to burden the Bitcoin activity with the 'work' required by their transaction data.

The BLN has, thus, promised to represent a solution to the Bitcoin scalability problem that does not sacrifice the key Bitcoin features which are 1) *decentralisation* (characterising its architecture, i.e. the number of

computers constituting the network), 2) its *political organisation* (i.e. the number of individuals controlling the network) and 3) its *wealth distribution* (i.e. the number of individuals owning the actual supply), while enhancing the circulation and the exchange of the native assets.

This chapter is devoted to verify if the promise has been fulfilled, by analysing the structure of the BLN over a period of 18 months, ranging from 12th January 2018 to 17th July 2019, across three different BLN representations: the daily snapshot one, the weekly snapshot one and the daily-block snapshot one, reviewing the data from Lin et al. (2020) and Vallarano et al. (2020).

5.2 The Bitcoin Lightning Network: basic statistics

As observed for the BANs and the BUNs considered in chapter 2, both the number of nodes N and the number of links L of the BLN increase steadily in time, while it becomes sparser. Interestingly, the evolution of the BLN link density seems to point out the presence of two regimes: as figure 24 shows, during the first phase, i.e. $N \lesssim 10^3$, L increases linearly as a function of N and the link density is well described by the functional dependence $d \sim N^{-1}$; afterwards, the link density decrease slows down, seemingly indicating that L has started to grow in a super-linear fashion with respect to N. This is confirmed by plotting the link density $d = \frac{2L}{N(N-1)}$ versus the number of nodes: the functional form $d \sim N^{-1}$ overlaps with the empirical trend up to the value $N \simeq 10^3$; afterwards, a different functional form appears (Lin et al. (2020)).

Let us now comment on the evolution of the cumulative density function (CDF) of the degrees, defined as

$$\text{CDF}(k) = \sum_{h \geq k} f(h) \tag{5.1}$$

where $f(h)$ is the fraction of nodes whose degree is h. Figure 25 shows the CDF for eight distinct snapshots, spanning the entire BLN history: as it can be appreciated, the degree distribution becomes broader as the

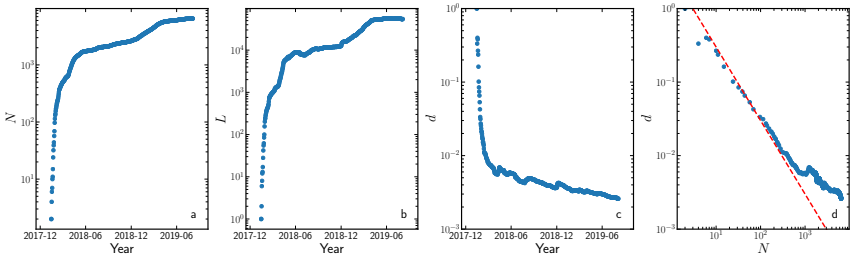

Figure 24: Evolution of the total number of nodes N, total number of links L and link density $d = \frac{2L}{N(N-1)}$ for the BLN (only the daily-block snapshot representation is considered here). As for the BANs and the BUNs considered in chapter 2, the position $d \sim N^{-1}$ well describes the link density dependence on N, for the snapshots for which $N \lesssim 10^3$. See also Vallarano et al. (2020).

BLN evolves. Overall, it resembles a power-law, although a divergence from this model clearly appears in the last snapshots (in fact, it bends in the initial and the final portions of the distribution).

5.3 The Bitcoin Lightning Network: mesoscale structure

Although blockchain-based systems are designed to get rid of the presence of a central authority that checks the validity of the exchanges between nodes - transactions, in the case of cryptocurrencies - and authorizes them, it can be shown that centralization may still be recovered at a purely structural level.

Indices measuring the centrality of a node aim at quantifying the importance of a node in a network, according to some specific topological property. Among the measures proposed so far, of particular relevance are the *degree centrality*, the *closeness centrality*, the *betweenness centrality* and the *eigenvector centrality*:

- *degree centrality*: the degree centrality k_i^c of node i is the number of its neighbours, normalized by the maximum number of neigh-

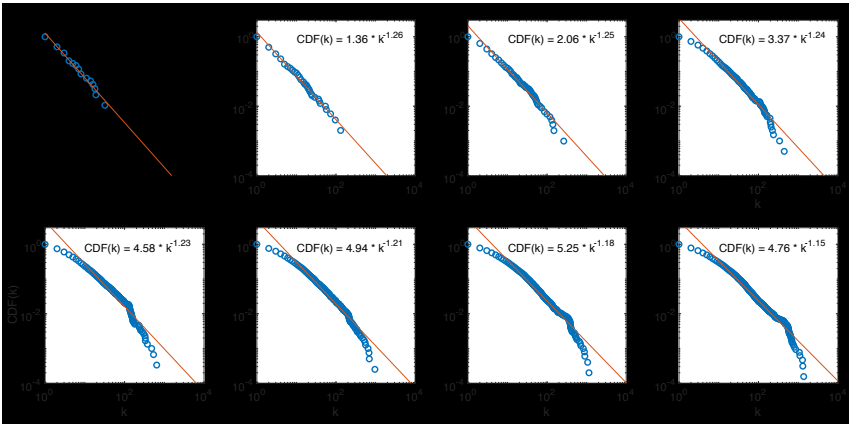

Figure 25: Evolution of the degree Cumulative Density Function for the snapshots whose LCC is characterized by a number of nodes amounting at 100, 500, 1.000, 2.000, 3.000, 4.000, 5.000 and 6.447. As the BLN evolves, the support of the distribution becomes broader, while it progressively deviates from a power-law.

bours, i.e. $N-1$ (see Newman (2018) and Latora et al. (2017)):

$$k_i^c = \frac{k_i}{N-1}; \tag{5.2}$$

- *closeness centrality*: the closeness centrality c_i^c of node i measures how 'close' it is to all the other nodes. Nodes are closer the fewer the number of links separating them (see Newman (2018) and Latora et al. (2017)):

$$c_i^c = \frac{N-1}{\sum_{j(\neq i)=1}^{N} d_{ij}} \tag{5.3}$$

where d_{ij} is the topological distance between nodes i and j, i.e. the length of the shortest path (in terms of number of links) connecting the two nodes, and the term $N-1$ normalizes it between 0 and 1. Clearly, if a node is connected to all the other nodes in the network, its topological distance from any of them is just 1;

- *betweenness centrality*: it measures how many times node i lies 'in between' the shortest paths connecting any two nodes of the network and is defined as

$$b_i^c = \sum_{s(\neq i)=1}^{N} \sum_{t(\neq i,s)=1}^{N} \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{5.4}$$

  where $\sigma_{st}(i)$ is the number of shortest paths between s and t passing trough i;

- *eigenvector centrality*: is defined via the spectral properties of the adjacency matrix of the network; in particular, e_i^c is the i-th element of the eigenvector corresponding to the largest eigenvalue of the binary adjacency matrix (whose existence is ensured by the Perron-Frobenius theorem). Large values of eigenvector centrality point out that the node is connected to other nodes whose eigenvector centrality is large well - in a sense, the 'well connected' ones (see Newman (2018) and Latora et al. (2017)). In this respect, its behaviour is similar to the PageRank centrality index.

While the analysis of the aforementioned measures is interesting *per se*, here we will study them to inspect the distribution of their values. To this aim, we will employ two measures we introduced in the previous chapter, i.e. the *Gini coefficient*

$$G_c = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |c_i - c_j|}{2N \sum_{i=1}^{N} c_i} \tag{5.5}$$

where $c_i = k_i^c, c_i^c, b_i^c, e_i^c$ (see Morgan (1962)) and the *centralization index*

$$C_c = \frac{\sum_{i=1}^{N} (c^* - c_i)}{\max\left\{\sum_{i=1}^{N} (c^* - c_i)\right\}} \tag{5.6}$$

where $c^* = \max\{c_i\}_{i=1}^{N}$ represents the empirical, maximum value of the chosen centrality measure and the denominator is computed over the benchmark graph, i.e. the one maximizing the sum at the numerator. As

shown in Lin et al. (2020), the reference network for the aforementioned centrality measures is the star graph[1]. More explicitly, it can be shown that

- *degree centralization index*:

$$C_{k^c} = \frac{\sum_i (k^* - k_i^c)}{N-2}; \tag{5.7}$$

- *closeness centralization index*:

$$C_{c^c} = \frac{\sum_i (c^* - c_i^c)}{(N-1)(N-2)/(2N-3)}; \tag{5.8}$$

- *betweenness centralization index*:

$$C_{b^c} = \frac{\sum_i (b^* - b_i^c)}{(N-1)^2(N-2)/2}; \tag{5.9}$$

- *eigenvector centralization index*:

$$C_{e^c} = \frac{\sum_i (e^* - e_i^c)}{(\sqrt{N-1}-1)(N-1)/(\sqrt{N-1}+N-1)}. \tag{5.10}$$

Figure 26 depicts the evolution of the Gini coefficient and of the centralization index: interestingly, while G_c increases for three measures out of four, pointing out that the values of centrality are more and more unevenly distributed, the evolution of the centralisation index indicates that the BLN is not evolving towards a star graph - indeed, a too simplistic picture (see Lin et al. (2020)). Notice that the flat trend of the closeness centrality can be explained by invoking the presence of nodes with large degree that ensure that the vast majority of nodes are, in turn, easily reachable.

In order to gain further insight into the BLN organization, it is interesting to benchmark the observations concerning the evolution of centrality and centralisation with the predictions, for the same quantities,

[1] While this is true for the degree closeness and betweenness centrality, this is not the case for the eigenvector centrality; still, it is kept for the sake of homogeneity with the other quantities.

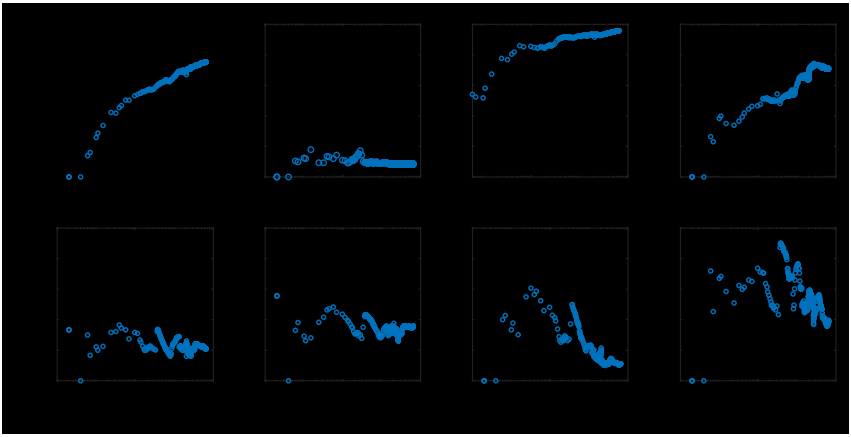

Figure 26: Evolution of the Gini coefficient and the centralization index for the four centrality measures chosen here, calculated on the daily-block snapshot representation of the BLN: G_c is characterised by a rising trend, irrespectively from the chosen indicator, pointing out that the values of centrality are increasingly unevenly distributed; on the other hand, the evolution of centralisation reveals that the picture provided by a star graph is too simple to faithfully represent the BLN structure. See also Lin et al. (2020).

output by the maximum-entropy null model known as *Undirected Binary Configuration Model* (UBCM - i.e. the undirected version of the DBCM, introduced in the previous chapter). To this aim, we have solved the UBCM by implementing the iterative, reduced algorithm

$$\begin{aligned} k_i(\mathbf{A}) \quad &= \quad \sum_{j(\neq i)=1}^{N} \frac{x_i x_j}{1+x_i x_j}, \; \forall \, i \\ \implies \quad x_k^{(n)} &= \frac{k(\mathbf{A})}{\sum_{k'} f(k') \left[\frac{x_{k'}^{(n-1)}}{1+x_k^{(n-1)} x_{k'}^{(n-1)}} \right] - \frac{x_k^{(n-1)}}{1+(x_k^2)^{(n-1)}}}, \; \forall \, k \end{aligned} \tag{5.11}$$

a choice allowing us to solve it within tens of seconds even for configurations with thousands of nodes (see Vallarano et al. (2021a) and Vallarano et al. (2021b)). Afterwards, we have explicitly sampled the ensembles of

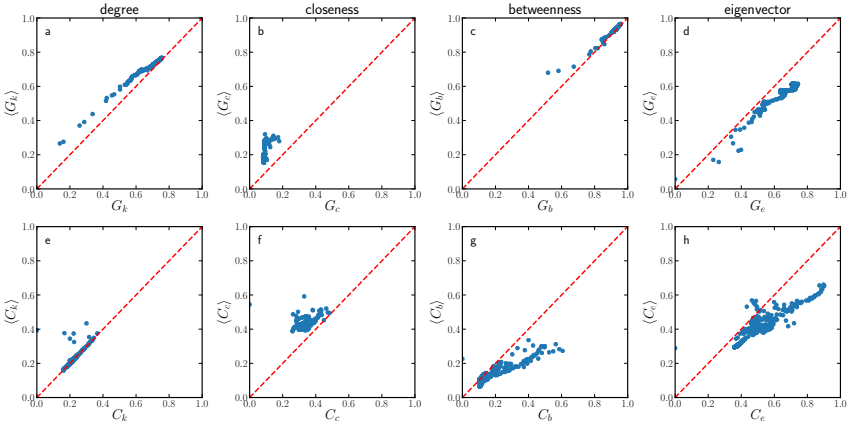

Figure 27: Top panels: comparison between the observed Gini index for the degree, closeness, betweenness and eigenvector centrality (x-axis) and their expected value, computed under the UBCM (y-axis) for the BLN daily-block snapshot representation. Bottom panels: comparison between the observed degree, closeness, betweenness and eigenvector centralisation and their expected value computed under the UBCM. Once the information contained into the degree sequence is properly accounted for, a (residual) tendency to centralisation is still visible. See also Vallarano et al. (2020).

networks induced by the UBCM (see Park et al. (2004) and Squartini et al. (2011)) and compared the (ensemble) average of the quantities of interest with the corresponding empirical values.

As figure 27 shows, this comparison reveals that the UBCM tends to overestimate the values of the Gini index for the degree, the closeness and the betweenness centrality and to underestimate its values for the eigenvector centrality. This seems to point out a non-trivial (i.e. not reproducible by just enforcing the degrees) tendency of well-connected nodes to establish connections among themselves. Moreover, these very connected nodes have nodes with smaller degree attached to them (see Lin et al., 2020), thus generating a disassortative structure that explains the less-than-expected level of unevenness characterising the other centrality measures: in fact, the nodes behaving as the 'leaves' of the hubs

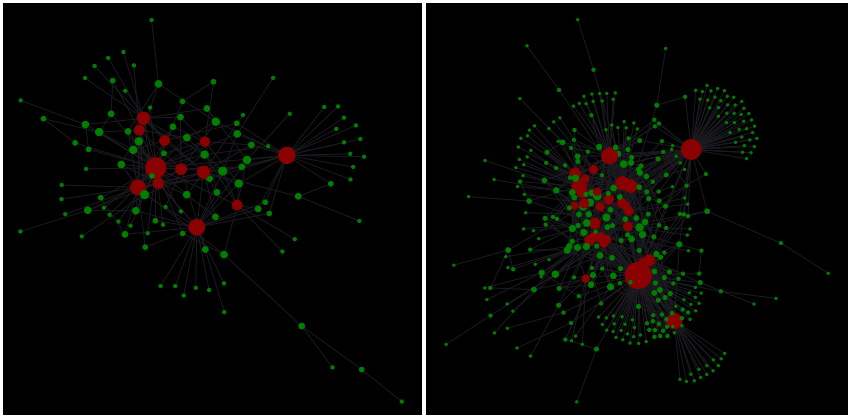

Figure 28: Core-periphery structure of the BLN daily-block snapshot representation on day 17 (left panel) and on day 35 (right panel), with core-nodes drawn in red and periphery-nodes drawn in green. See also Vallarano et al. (2020).

basically have the same values of degree, closeness and betweenness centrality.

For what concerns the analysis of the centralisation indices, figure 27 shows that the UBCM underestimates both the betweenness and the eigenvector centralisation indices: in other words, a tendency to centralisation 'survives' even after the information encoded into the degrees is properly accounted for, letting the picture of a network characterised by some kind of more-than-expected 'star-likeness' emerge.

This observation can be better formalised by analysing the BLN mesoscale structure via the optimization of the surprise score function, introduced in the previous chapter: as observed for our BUNs, a core-periphery structural organization, whose statistical significance increases over time, indeed emerges (see Lin et al. (2020) and figure 28).

As the network analysis reveals, the BLN is evolving towards an increasingly centralised architecture (in particular, a core-periphery one) where many star-like sub-structures, whose centers coincide with the 'centrality hubs' revealed by the Gini coefficient, co-exist (Lin et al. (2020)).

These hubs act as channel-switching nodes and seem to emerge as an unavoidable consequence of the way BLN is designed. As a route through the network must be found and longer routes are more expensive (fees are present for the 'gateway service' provided by intermediate nodes), any two BLN users will search for a short(est) path: at the same time, nodes (which can only create channels based on local information) have the incentive to become as central as possible, within the BLN, in order to maximize the transaction fees they may earn. Hubs may, thus, have emerged as a consequence of the collective action of users following the two aforementioned behaviours - and, from this perspective, it is not surprising that central nodes have been observed since the very beginning of the BLN history.

For what concerns hubs interconnectedness, then, previous results have shown that mechanisms of centrality-maximizing agents yield a core-periphery structure (regardless of the notion of centrality the agents attempt to maximize), an evidence indicating that the presence of both topological signatures can be compactly inspected by studying (the evolution) of eigenvector centrality (see Konig et al. (2010) and Konig et al. (2014)). As a last observation, we also notice that the presence of 'centrality hubs' seems to be at the origin of another structural BLN peculiarity, i.e. its small-world -ness, a feature already revealed by previous studies (see Rohrer et al. (2019)).

The tendency of the BLN architecture to become 'less distributed' has the undesirable consequence of making the BLN increasingly less resilient against random failures, malicious attacks (e.g. the so-called 'split ones'), etc.

5.4 The Bitcoin Lightning Network: a quick look at its weighted structure

The empirical analysis of the BLN weighted structure can be inspected by plotting the CDF of the weights and the strengths for the same snapshots considered for the degrees. As figure 29 reveals, the CDF of the weights does not resemble *entirely* a power-law: the initial part, in fact, seems

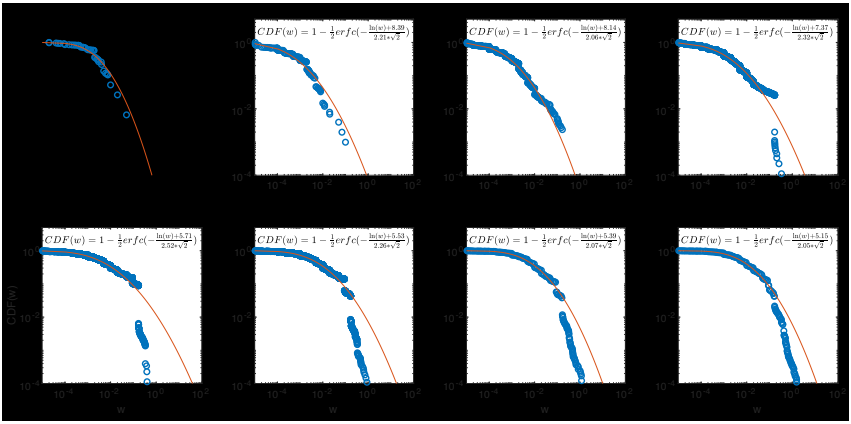

Figure 29: Cumulative Density Function of the weights for the snapshots whose LCC is characterized by a number of nodes amounting at 100, 500, 1.000, 2.000, 3.000, 4.000, 5.000 and 6.447. While its initial part seems to obey a log-normal, the last one appears as more similar to a power law; moreover, its support has remained quite constant throughout the entire history of the BLN. See also Lin et al. (2020).

to obey a log-normal distribution - while the power-law behaviour appears in the last one; moreover, its support has remained quite constant throughout the entire BLN history. For what concerns the CDF of the strengths, instead, its agreement with a log-normal is remarkable; moreover, its support has broadened during the last snapshots, as figure 30 reveals.

As already pointed out in Lin et al. (2020), unevenness affects the distribution of weighted quantities as well. This is the case of the *total amount of exchanged bitcoins* and of the *strength sequence*. For what concerns the first quantity, it grows approximately with the square of the network size; still, it has been found that the percentage of nodes holding the 80%, 90%, 95% and 99% of the total number of bitcoins at stake in the network (intended as the fraction of top nodes whose total strength amounts at the aforementioned percentages) is (about) 10%, 20%, 30% and 50% of the total.

For what concerns the second quantity, the evolution of the Gini co-

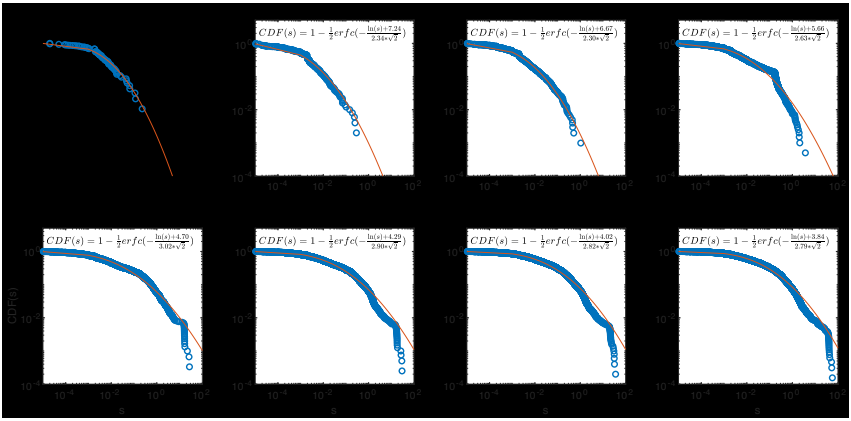

Figure 30: Cumulative Density Function of the strengths for the snapshots whose LCC is characterized by a number of nodes amounting at 100, 500, 1.000, 2.000, 3.000, 4.000, 5.000 and 6.447. Its agreement with a log-normal is remarkable; moreover, its support has remained quite constant for a large portion of the BLN history, while it has broadened in the last snapshots. See also Lin et al. (2020).

efficient

$$G_s = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |s_i - s_j|}{2N \sum_{i=1}^{N} s_i}, \tag{5.12}$$

quantifying the unevenness of the distribution of strengths, reveals it to rise in an almost monotonic fashion throughout the entire BLN history, its average value amounting at $\simeq 0.88$ for the daily-block snapshot representation.

Chapter 6

Solving null models on very large networks

Different frameworks exist to model complex networks. Our approach is based on the Exponential Random Graphs one, that has gained increasing popularity over the years. Rooted into statistical physics, the ERGMs workflow is defined by two subsequent optimization steps: the first one concerns the maximization of Shannon entropy and identifies the functional form of the ensemble probability distribution; the second one concerns the maximization of the likelihood function induced by the latter and leads to its numerical determination. This second step translates into the resolution of a system of $O(N)$ non-linear, coupled equations, a problem that is affected by three main issues, i.e. accuracy, speed and scalability. The present chapter, whose content partly overlaps with the one of the paper authored by Vallarano et al. (2021b), is devoted to address these issues.

6.1 Exponential Random Graph Models

Let us, first, introduce the framework defining the Exponential Random Graph Models (ERGMs hereby). It is defined by the maximization of Shannon entropy, i.e.

$$S = -\sum_{\mathbf{G}\in\mathcal{G}} P(\mathbf{G}) \ln P(\mathbf{G}) \tag{6.1}$$

constrained to satisfy a set of quantities assumed to represent a meaningful piece of information to be preserved. This requirement can be formally imposed by writing down the Lagrangean function

$$\mathcal{L} = S - \sum_{i=0}^{M} \left[\sum_{\mathbf{G}\in\mathcal{G}} P(\mathbf{G}) x_i(\mathbf{G}) - \langle x_i \rangle \right] \tag{6.2}$$

where M indicates the total number of imposed constraints and $x_0 \equiv \langle x_0 \rangle \equiv 1$ encodes the normalization condition. $P(\mathbf{G})$ is the probability of the configuration $\mathbf{G} \in \mathcal{G}$ where the ensemble $\mathcal{G}$ can be defined as the set of (binary or weighted, undirected or directed) networks with the same number of nodes (say, N). The number of links is, of course, allowed to vary: for binary networks, it varies from 0 to the maximum (i.e. $\frac{N(N-1)}{2}$, in case of undirected networks and $N(N-1)$, in case of directed networks).

The constrained maximization of Shannon entropy allows us to define a probability distribution such that the expected value, over the ensemble, of the set of quantities $\{x_i\}_{i=1}^{M}$ is fixed. Following Park et al. (2004), the ensemble distribution is defined as:

$$P(\mathbf{G}|\underline{\theta}) = \frac{e^{-H(\mathbf{G},\underline{\theta})}}{Z}, \quad \forall\, \mathbf{G} \in \mathcal{G}; \tag{6.3}$$

notice that the so-called *network Hamiltonian*

$$H(\mathbf{G}, \underline{\theta}) = \sum_{i=1}^{M} \theta_i x_i(\mathbf{G}) \tag{6.4}$$

sums up the constraints, imposed via the vector of parameters $\underline{\theta}$ each of which controls for the value of one of the measures $\{x_i\}_{i=1}^{M}$; Z, instead, is the *partition function*, defined as

$$Z = \sum_{\mathbf{G}} e^{-H(\mathbf{G},\underline{\theta})} \tag{6.5}$$

and ensuring that $P(\mathbf{G})$ is correctly normalized on $\mathcal{G}$. Naturally,

$$\sum_{\mathbf{G}} P(\mathbf{G}) x_i(\mathbf{G}) = \langle x_i \rangle, \quad i = 1 \dots M. \tag{6.6}$$

While the functional form of $P(\mathbf{G})$ has been determined by maximizing Shannon entropy, its numerical determination requires a different principle. Following Garlaschelli et al. (2008) and Squartini et al. (2011), one can invoke the likelihood maximization principle, that formalizes the requirement that the probability of observing the actual network configuration $\mathbf{G}^*$ must be maximum. This translates into solving the problem

$$\max_{\underline{\theta}} \lambda(\mathbf{G}^*, \underline{\theta}) = \max_{\underline{\theta}} \log P(\mathbf{G}^* | \underline{\theta}); \tag{6.7}$$

or, equivalently, the following set of equations

$$\frac{\partial \lambda(\mathbf{G}^*, \underline{\theta})}{\partial \theta_i} = 0, \quad i = 1 \dots M \tag{6.8}$$

also known as *first-order conditions*. Now, upon substituting the expression defined by equation 6.3 into equation 6.7, one finds that the likelihood maximization problem translates into the resolution of the system of equations

$$\langle x_i \rangle = x_i^*, \quad i = 1 \dots M \tag{6.9}$$

with M being the number of constraints, i.e. the number of equations to solve as well as the dimension of the search space for the optimization problem 6.7.

In what follows, we are going to instantiate the framework we have described so far by considering the simplest, yet not trivial, set of constraints, i.e. the degree sequence for undirected networks.

6.2 The Undirected Binary Configuration Model

The *Undirected Binary Configuration Model* (UBCM) is defined by constraining the degree sequence $\{k_i\}_{i=1}^N$ for binary, undirected networks. The model Hamiltonian reads

$$H_{\text{UBCM}}(\mathbf{A}, \underline{\theta}) = \sum_{i=1}^{N} \theta_i k_i(\mathbf{A}) \tag{6.10}$$

and induces the probability distribution

$$\begin{aligned} P_{\text{UBCM}}(\mathbf{A}|\underline{\theta}) &= \frac{e^{-\sum_{i=1}^{N} \theta_i k_i(\mathbf{A})}}{\sum_{\mathbf{A}} e^{-\sum_{i=1}^{N} \theta_i k_i(\mathbf{A})}} = \\ &= \frac{e^{-\sum_{i=1}^{N} \sum_{j(>i)=1}^{N} (\theta_i + \theta_j) a_{ij}}}{\sum_{\mathbf{A}} e^{-\sum_{i=1}^{N} \sum_{j(>i)=1}^{N} (\theta_i + \theta_j) a_{ij}}} = \\ &= \frac{\prod_{i=1}^{N} \prod_{j(>i)=1}^{N} e^{-(\theta_i + \theta_j) a_{ij}}}{\prod_{i=1}^{N} \prod_{j(>i)=1}^{N} \sum_{a_{ij} \in \{0,1\}} e^{-(\theta_i + \theta_j) a_{ij}}} = \\ &= \prod_{i=1}^{N} \prod_{j(>i)=1}^{N} \frac{e^{-(\theta_i + \theta_j) a_{ij}}}{1 + e^{-(\theta_i + \theta_j)}}; \end{aligned} \tag{6.11}$$

upon renaming $x_i \equiv e^{-\theta_i}$ one finds the more readable result

$$P_{\text{UBCM}}(\mathbf{A}|\underline{\theta}) = \prod_{i=1}^{N} \prod_{j(>i)=1}^{N} p_{ij}^{a_{ij}} (1 - p_{ij})^{1 - a_{ij}} = \prod_{i=1}^{N} x_i^{k_i(\mathbf{A})} \prod_{j(>i)=1}^{N} \frac{1}{1 + x_i x_j} \tag{6.12}$$

where we have posed

$$p_{ij} \equiv \frac{x_i x_j}{1 + x_i x_j}, \quad \forall\, i < j; \tag{6.13}$$

the UBCM probability factorizes into a product of coefficients, the generic one p_{ij} describing the probability that a link exists between nodes i and j. Notice that we haven't required link independence at any point: it is one of the outcomes of the UBCM.

Using the second expression in equation (6.12), one finds that the likelihood optimization problem defined by equation (6.7) now reads

$$\max_{\underline{x} \in \mathbb{R}_+^N} \left\{ \sum_{i=1}^{N} k_i(\mathbf{A}^*) \ln x_i - \sum_{i=1}^{N} \sum_{j(>i)=1}^{N} \ln(1 + x_i x_j) \right\} \tag{6.14}$$

the search space being defined by the position $x_i \equiv e^{-\theta_i} > 0, \forall\, i$ - obviously, we have explicitly excluded the case in which isolated nodes are present; in fact, a degree equal to zero induces a variable which is zero as well, i.e. $k_i = 0 \Longrightarrow x_i = 0$. The optimization problem to solve becomes

$$k_i(\mathbf{A}^*) = \sum_{j(\neq i)=1}^{N} \underbrace{\frac{x_i x_j}{1 + x_i x_j}}_{p_{ij}} = \langle k_i \rangle, \quad i = 1 \dots M \tag{6.15}$$

and tells us that the constraints defining the UBCM ensemble are satisfied on average (see Garlaschelli et al. (2008) and Squartini et al. (2011)).

The system above consists of N equations, i.e. one per node. When very large networks are considered, solving it may be unfeasible. For this reason, we need to find a way to reduce its computational complexity. In order to do so, we build upon a suggestion originally proposed in Garlaschelli et al. (2008), moving from the evidence that a sum of increasing, monotonic functions appears in equation (6.7); hence, only the distinct values of the degrees actually matter and reduction is induced by a shrinkage of the number of equations to solve.

Our strategy is that of rewriting H_{UBCM} in such a way that it doesn't depend on the full set of degrees but only on the reduced one. In order to do so, we need to prove that the idea behind reduction indeed works. Let's start with the following lemma:

Lemma 1. *Given the optimization problem 6.14, $k_i = k_j \Leftrightarrow x_i^* = x_j^*$.*

Proof. The implication $k_i = k_j \Leftarrow x_i^* = x_j^*$ is immediate. Let us now focus on the implication $k_i = k_j \Rightarrow x_i^* = x_j^*$. From equation (6.14) we know the solution $\underline{x}^*$ should satisfy the first order conditions above. Hence,

$$
\begin{aligned}
k_i = k_l \Longrightarrow \sum_{j(\neq i)=1}^{N} \frac{x_i x_j}{1 + x_i x_j} &= \sum_{j(\neq l)=1}^{N} \frac{x_l x_j}{1 + x_l x_j} \\
\sum_{j(\neq i,l)=1}^{N} \frac{x_i x_j}{1 + x_i x_j} + \frac{x_i x_l}{1 + x_i x_l} &= \sum_{j(\neq i,l)=1}^{N} \frac{x_l x_j}{1 + x_l x_j} + \frac{x_i x_l}{1 + x_i x_l} \\
\sum_{j(\neq i,l)=1}^{N} \left(\frac{x_i x_j}{1 + x_i x_j} - \frac{x_j x_l}{1 + x_j x_l} \right) &= 0 \\
\sum_{j(\neq i,l)=1}^{N} x_j \frac{x_i(1 + x_l x_j) - x_l(1 + x_i x_j)}{(1 + x_i x_j)(1 + x_l x_j)} &= 0 \\
(x_i - x_l) \sum_{j(\neq i,l)=1}^{N} \frac{x_j}{(1 + x_i x_j)(1 + x_l x_j)} &= 0; \qquad (6.16)
\end{aligned}
$$

from the last equality and the fact that $\frac{x_j}{(1+x_i x_j)(1+x_l x_j)} > 0$ for at least one j (or the problem would be a trivial one), it follows that $x_i = x_j$. $\square$

Lemma 1 confirms the intuition in Garlaschelli et al. (2008): the solution space of problem 6.14 is actually smaller than $\mathbb{R}_+^N$. Let us now exploit this fact to compute $\underline{x}$ in a faster, more efficient way. First, let us define a binary relation $\sim$ on the set of nodes $\mathcal{N}$ of our graph stating that, given $i, j \in \mathcal{N}$, $i \sim j \iff k_i = k_j$ and having the following properties:

1. $\sim$ is *reflexive*: $i \sim i$ because $k_i = k_i$;
2. $\sim$ is *symmetric*: $k_i = k_j$ implies that $k_j = k_i$ as well;
3. $\sim$ is *transitive*: if $k_i = k_j$ and $k_j = k_h$, then $k_i = k_h$.

Then, $\sim$ is an *equivalence relation* and we can define *equivalence classes*. An equivalence class is a subset $Y \subseteq \mathcal{N}$ such that $\forall\, i, j \in Y$, then $i \sim j$; moreover, if $h \notin Y$, then $i \nsim h$. We indicate the equivalent classes as

$$
[\alpha] = [k_\alpha] = \{ i \in \mathcal{N} \quad \text{s.t.} \quad k_i = k_\alpha \}, \quad \forall\, \alpha \in \mathcal{N} \qquad (6.17)
$$

In other words, all equivalent nodes (under $\sim$) are in the same class - and only them. The collection of all equivalence classes is called the *quotient set* and it is denoted by $\mathcal{N}/\sim$. Now, from the fundamental theorem on equivalence relations (see Wallace (2012), page 31), we know that the quotient set generated by $\sim$ induces a partition of the nodes; then, from $\sim$ we can define the projection $\pi : \mathcal{N} \to \mathcal{N}/\sim$ which maps each node in its equivalence class, i.e. $\pi(i) = [i]$. At this point, we can define the subspace $X^\pi \subset \mathbb{R}_+^N$ such that $\forall\, \underline{x} \in X^\pi$, $x_i = x_j \iff \pi(i) = \pi(j)$. Lemma 1 proved that the solution $\underline{x}^*$ of the problem 6.7 is in X^π.

Let us define the map $\phi : X^\pi \subset \mathbb{R}_+^{N^{red}} \to \mathbb{R}_+^{N^{red}}$, where $N^{red} = |\mathcal{N}/\sim|$ is the cardinality of the quotient set (i.e. the number of equivalence classes), such that

$$\phi(\underline{x}) = (\phi_1(\underline{x}) \dots \phi_{N^{red}}(\underline{x})) \quad \text{and} \quad \phi_i(\underline{x}) = x_{\pi(i)}; \tag{6.18}$$

then, ϕ is well defined only on X^π and it's invertible on it. Actually, we can prove (proof is omitted) that ϕ is an omeomorphism. Finally, we can define $\lambda^{red}_{\text{UBCM}} : \mathbb{R}^{N^{red}} \to \mathbb{R}$ as the reduced log-likelihood

$$\lambda^{red}_{\text{UBCM}}(\mathbf{A}, \underline{x}) \equiv \lambda_{\text{UBCM}|X^\pi}(\mathbf{A}, \phi^{-1}(\underline{x})), \quad \forall\, \underline{x} \in \mathbb{R}_+^{N^{red}} \tag{6.19}$$

Why is $\lambda^{red}_{\text{UBCM}}$ so important? Because we know that the solution to the original problem $\underline{x}^*$ is in the quotient set X^π and the new log-likelihood function we wrote is defined on the same quotient set and it assumes the same values of the original log-likelihood on it (equation 6.19). The strategy is, then, solving the optimization problem induced by the reduced log-likelihood on $\mathbb{R}^{N^{red}}$ (a space with less dimensions is usually associated with an easier problem to solve) and then recover the (full) solution associated with the original problem 6.14, via ϕ.

The reduced optimization problem reads

$$\max_{\underline{x} \in \mathbb{R}_+^{N^{red}}} \lambda^{red}_{\text{UBCM}}(\mathbf{A}^*, \underline{x}) \tag{6.20}$$

where

$$\begin{aligned}\lambda^{red}_{\text{UBCM}}(\mathbf{A}^*, \underline{x}) &= \lambda_{\text{UBCM}|X^\pi}(\mathbf{A}^*, \phi^{-1}(\underline{x})) \\ &= \sum_\alpha |[k_\alpha]| k_\alpha \log x_\alpha \\ &\quad - \sum_\alpha \sum_{\beta(\geq\alpha)} |[k_\alpha]|(|[k_\beta]| - \mathbb{1}_{\alpha\neq\beta}) \log(1 + x_\alpha x_\beta)\end{aligned} \tag{6.21}$$

where $|\cdot|$ is the cardinality of a set. Applying the first order conditions to equation (6.21), we obtain a reduced version of the first order conditions seen in equation (6.15)

$$0 = \overbrace{\frac{k_\gamma |[k_\gamma]|}{x_\gamma} - \binom{|[k_\gamma]|}{2} \frac{2x_\gamma}{1+x_\gamma^2} - \sum_{\beta(\neq\gamma)} |[k_\gamma]||[k_\beta]| \frac{x_\beta}{1+x_\beta x_\gamma}}^{\frac{\partial \lambda_{\text{UBCM}}}{\partial x_\gamma}} \tag{6.22}$$

that can be rewritten as

$$k_\gamma = \sum_{\beta(\neq\gamma)} |[k_\beta]| \frac{x_\beta x_\gamma}{1+x_\beta x_\gamma} + (|[k_\gamma]| - 1) \frac{x_\gamma^2}{1+x_\gamma^2}. \tag{6.23}$$

6.3 The Directed Binary Configuration Model

A straightforward generalization of the UBCM for binary, directed networks is the *Directed Binary Configuration Model* (DBCM). The constraints are, now, the in- and out-degrees of each node, i.e. $\{k_i^{out}\}_{i=1}^N$ and $\{k_i^{in}\}_{i=1}^N$. The Hamiltonian is defined as:

$$H_{\text{DBCM}}(\mathbf{A}, \underline{\theta}) = \sum_{i=1}^N \left(\theta_i^{in} k_i^{in}(\mathbf{A}) + \theta_i^{out} k_i^{out}(\mathbf{A})\right) \tag{6.24}$$

and $\theta \equiv (\underline{\theta}^{in}, \underline{\theta}^{out})$. As for the UBCM, the probability of generating a graph from the DBCM ensemble reads

$$
\begin{aligned}
P_{\text{DBCM}}(\mathbf{A}|\underline{\theta}) &= \frac{e^{-\sum_{i=1}^{N}\left(\theta_i^{in}k_i^{in}(\mathbf{A})+\theta_i^{out}k_i^{out}(\mathbf{A})\right)}}{\sum_{\mathbf{A}} e^{-\sum_{i=1}^{N}\left(\theta_i^{in}k_i^{in}(\mathbf{A})+\theta_i^{out}k_i^{out}(\mathbf{A})\right)}} \\
&= \frac{e^{-\sum_{i=1}^{N}\sum_{j(\neq i)=1}^{N}(\theta_i^{out}+\theta_j^{in})a_{ij}}}{\sum_{\mathbf{A}} e^{-\sum_{i=1}^{N}\sum_{j(\neq i)=1}^{N}(\theta_i^{out}+\theta_j^{in})a_{ij}}} \\
&= \frac{\prod_{i=1}^{N}\prod_{j(\neq i)=1}^{N} e^{-(\theta_i^{out}+\theta_j^{in})a_{ij}}}{\prod_{i=1}^{N}\prod_{j(\neq i)=1}^{N}\sum_{a_{ij}\in\{0,1\}} e^{-(\theta_i^{out}+\theta_j^{in})a_{ij}}} \\
&= \prod_{i=1}^{N}\prod_{j(\neq i)=1}^{N}\frac{e^{-(\theta_i^{out}+\theta_j^{in})a_{ij}}}{1+e^{-(\theta_i^{out}+\theta_j^{in})}}; \qquad (6.25)
\end{aligned}
$$

upon renaming $x_i \equiv e^{-\theta_i^{out}} > 0$ and $y \equiv e^{-\theta_i^{in}} > 0$, we can write

$$
\begin{aligned}
P_{\text{DBCM}}(\mathbf{A}|\underline{\theta}) &= \prod_{i=1}^{N}\prod_{j(\neq i)=1}^{N} p_{ij}^{a_{ij}}(1-p_{ij})^{1-a_{ij}} \\
&= \prod_{i=1}^{N} x_i^{k_i^{out}(\mathbf{A})} y_i^{k_i^{in}(\mathbf{A})} \prod_{i=1}^{N}\prod_{j(\neq i)=1}^{N}\frac{1}{1+x_i y_j}; \qquad (6.26)
\end{aligned}
$$

where we have posed $p_{ij} \equiv \frac{x_i y_j}{1+x_i y_j}$, $\forall\, i \neq j$. As for the UBCM, the probability of a graph under the DBCM can be factorized as a product of pair-specific probability coefficients. From the Hamiltonian in equation (6.25), the optimization problem becomes

$$
\max_{\underline{x},\underline{y}\in\mathbb{R}_+^{2N}}\left\{\sum_{i=1}^{N}\left(k_i^{in}(\mathbf{A}^*)\log x_i + k_i^{out}(\mathbf{A}^*)\log y_i\right) - \sum_{i=1}^{N}\sum_{j(\neq i)=1}^{N}\log\left(1+x_i y_j\right)\right\}; \qquad (6.27)
$$

analogously to the undirected case, the first-order conditions read

$$k_i^{out}(\mathbf{A}^*) = \sum_{j(\neq i)=1}^{N} \frac{x_i y_j}{1 + x_i y_j} \tag{6.28}$$

$$k_i^{in}(\mathbf{A}^*) = \sum_{j(\neq i)=1}^{N} \frac{x_j y_i}{1 + y_i x_j} \tag{6.29}$$

that we can reduce by proving a lemma which is analogous to the one we proved for the UBCM.

Lemma 2. *Given the optimization problem 6.27, $k_i^{out} = k_l^{out}$ and $k_i^{out} = k_l^{out}$ $\iff x_i^* = x_j^*$ and $y_i^* = y_j^*$.*

Proof. The implication $k_i^{out} = k_l^{out}$ and $k_i^{out} = k_l^{out} \impliedby x_i^* = x_j^*$ and $y_i^* = y_j^*$ is immediate. Let us now focus on the implication $k_i^{out} = k_l^{out}$ $\implies x_i^* = x_j^*$ and $y_i^* = y_j^*$. For hypothesis, let us consider two nodes i and l such that $k_i^{out} = k_l^{out}$ and $k_i^{in} = k_l^{in}$; now, from equation (6.29)

$$\begin{aligned}
\sum_{j(\neq i)=1}^{N} \frac{y_i x_j}{1 + y_i x_j} - \sum_{j(\neq l)=1}^{N} \frac{y_l x_j}{1 + y_l x_j} &= 0 \\
\sum_{j(\neq i,l)=1}^{N} \left(\frac{y_i x_j}{1 + y_i x_j} + \frac{y_i x_l}{1 + y_i x_l} \right) - \sum_{j(\neq i,l)=1}^{N} \left(\frac{y_l x_j}{1 + y_l x_j} + \frac{y_l x_i}{1 + y_l x_i} \right) &= 0 \\
\sum_{j(\neq i,l)=1}^{N} \left(\frac{y_i x_j}{1 + y_i x_j} - \frac{y_l x_j}{1 + y_l x_j} \right) + \frac{y_i x_l}{1 + y_i x_l} - \frac{y_l x_i}{1 + y_l x_i} &= 0 \\
(y_i - y_l) \sum_{j(\neq i,l)=1}^{N} \frac{x_j}{(1 + y_i x_j)(1 + y_l x_j)} - \left(\frac{y_l x_i}{1 + y_l x_i} - \frac{y_i x_l}{1 + y_i x_l} \right) &= 0
\end{aligned}$$

or, in a more compact notation,

$$(y_i - y_l)K - \Delta = 0 \tag{6.30}$$

where

$$K = \sum_{j(\neq i,l)=1}^{N} \frac{x_j}{(1+y_i x_j)(1+y_l x_j)} > 0 \tag{6.31}$$

$$\Delta = \frac{y_l x_i}{1+y_l x_i} - \frac{y_i x_l}{1+y_i x_l}; \tag{6.32}$$

in the same fashion, from equation (6.28) we obtain

$$\begin{aligned} \sum_{j(\neq i)=1}^{N} \frac{x_i y_j}{1+x_i y_j} - \sum_{j(\neq l)=1}^{N} \frac{x_l y_j}{1+x_l y_j} &= 0 \\ \sum_{j(\neq i,l)=1}^{N} \left(\frac{x_i y_j}{1+x_i y_j} - \frac{x_l y_j}{1+x_l y_j} \right) + \left(\frac{x_i y_l}{1+x_i y_l} - \frac{x_l y_i}{1+x_l y_i} \right) &= 0 \\ (x_i - x_l)K' + \Delta &= 0 \end{aligned} \tag{6.33}$$

where, now

$$K' = \sum_{j(\neq i,l)=1}^{N} \frac{y_j}{(1+x_i y_j)(1+x_l y_j)} > 0; \tag{6.34}$$

if we sum up equation (6.30) and equation (6.33) we obtain:

$$(x_i - x_l)K - \Delta + (y_i - y_l)K' + \Delta = 0 \tag{6.35}$$

i.e.

$$(x_i - x_l)K = -(y_i - y_l)K'. \tag{6.36}$$

We want to prove that $x_i = x_l$ and $y_i = y_l$: let us proceed by contradiction and assume $x_i \neq x_l$: then, either $x_i > x_l$ or $x_i < x_l$. Let us pick the case $x_i > x_l$ (the other case is analogous): now, by equation (6.36)

$$x_i > x_l \iff y_i < y_l; \tag{6.37}$$

let us now show that, from equation (6.37), it follows that $k_i^{in} \neq k_l^{in}$ (to be precise, $k_i^{in} > k_l^{in}$), i.e. our contradiction. First, notice that equation (6.37) leads to the result $x_i y_l > x_l y_i$; then, consider the monotonically increasing function $f(z) = \frac{z}{1+z}$, allowing us to write

$$\Delta = \frac{x_i y_l}{1 + x_i y_l} - \frac{x_l y_i}{1 + x_l y_i} = f(x_i y_l) - f(x_l y_i) > 0 \tag{6.38}$$

in turn, leading to the expression

$$k_i^{in} - k_l^{in} = \underbrace{(x_i - x_l)}_{>0} \underbrace{K'}_{>0} + \underbrace{\Delta}_{>0} > 0 \tag{6.39}$$

which is the contradiction we searched for. □

Now, we define the equivalence relation underlying the reduction strategy for the DBCM. As for the UBCM, let us define the binary relation $\sim$ on the set of nodes $\mathcal{N}$, stating that, given $i, j \in \mathcal{N}$, $i \sim j \iff k_i^{out} = k_j^{out}$ and $k_i^{in} = k_j^{in}$ and having the following properties:

1. $\sim$ is *reflexive*: $i \sim i$ because $k_i^{out} = k_i^{out}$ and $k_i^{in} = k_i^{in}$;
2. $\sim$ is *symmetric*: $k_i^{out} = k_j^{out}$ implies that $k_j^{out} = k_i^{out}$ as well (and analogously for the in-degree);
3. $\sim$ is *transitive*: if $k_i^{out} = k_j^{out}$ and $k_j^{out} = k_h^{out}$ then $k_i^{out} = k_h^{out}$ (and analogously for the in-degree).

Then, $\sim$ is an *equivalence relation* and we can define *equivalence classes*. We indicate the equivalent classes as

$$[\alpha] = [k_\alpha] = \left\{ i \in \mathcal{N} \quad \text{s.t.} \quad k_i^{out} = k_\alpha^{out} \quad \text{and} \quad k_i^{in} = k_\alpha^{in} \right\} \quad \forall\, \alpha \in \mathcal{N}; \tag{6.40}$$

the *quotient set* is denoted by $\mathcal{N}/\sim$ and it's a partition of the set of nodes $\mathcal{N}$. From $\sim$, we define the projection $\pi : \mathcal{N} \to \mathcal{N}/\sim$ that maps each node in its equivalence class: $\pi(i) = [i]$. As for the UBCM, we define the sub-space $X^\pi \subset \mathbb{R}^{2N}$ such that $\forall\, (\underline{x}, \underline{y}) \in X^\pi$, $x_i = x_j$ and $y_i = y_j \iff \pi(i) = \pi(j)$. Lemma 2 proved that the solution $(\underline{x}^*, \underline{y}^*)$ of problem 6.7 is in X^π.

Let us define the map $\phi : X^\pi \subset \mathbb{R}^{2N^{red}} \to \mathbb{R}^{2N^{red}}$, where $N^{red} = |\mathcal{N}/\sim|$ is the cardinality of the quotient set (i.e. the number of equivalence classes), such that

$$\phi(\underline{x},\underline{y}) = (\phi_1^x(\underline{x},\underline{y})\dots\phi_{N^{red}}^x(\underline{x},\underline{y}),\phi_1^y(\underline{x},\underline{y})\dots\phi_{N^{red}}^y(\underline{x},\underline{y})) \tag{6.41}$$

and

$$\phi_i^x(\underline{x},\underline{y}) = x_{\pi(i)}, \quad \phi_i^y(\underline{x},\underline{y}) = y_{\pi(i)}; \tag{6.42}$$

observe that ϕ is well-defined only on X^π, where it is also invertible. Moreover, it can be proven that ϕ is an omeomorphism. As in the undirected case, we can define $\lambda_{\text{DBCM}}^{red} : \mathbb{R}^{2N^{red}} \to \mathbb{R}$ as the reduced log-likelihood such that:

$$\lambda_{\text{DBCM}}^{red}(\mathbf{A}^*,\underline{x},\underline{y}) \equiv \lambda_{\text{DBCM}|X^\pi}(\mathbf{A}^*,\phi^{-1}(\underline{x},\underline{y})), \quad \forall \underline{x},\underline{y} \in \mathbb{R}^{2N^{red}}; \tag{6.43}$$

following the same strategy outlined for the UBCM, we solve the optimization problem for the reduced log-likelihood on $\mathbb{R}^{2N^{red}}$ and, then, recover the full solution of the original problem 6.14, via ϕ.

The reduced version of the log-likelihood defined by equation (6.27) becomes

$$\begin{aligned}\lambda_{\text{DBCM}}^{red}(\mathbf{A}^*,\underline{x}) &= \sum_\alpha |[k_\alpha]|(k_\alpha^{out}\log x_\alpha + k_\alpha^{in}\log y_\alpha) + \\ &\quad -\sum_\alpha\sum_\beta |[k_\alpha]|(|[k_\beta]| - (|[k_\beta]|-1)\mathbb{1}_{\alpha\neq\beta})\log(1+x_\alpha y_\beta)\end{aligned} \tag{6.44}$$

applying the first-order conditions to which we found

$$0 = \overbrace{\frac{k_\gamma^{out}|[k_\gamma]|}{x_\gamma} - \sum_{\beta\neq\gamma}\left(|[k_\gamma]||[k_\beta]|\frac{y_\beta}{1+x_\gamma y_\beta}\right) - |[k_\gamma]|(|[k_\gamma]|-1)\frac{y_\gamma}{1+x_\gamma y_\gamma}}^{\frac{\partial\lambda_{\text{DBCM}}^{red}}{\partial x_\gamma}} \tag{6.45}$$

a condition that translates into

$$k_\gamma^{out} = \sum_{\beta(\neq\gamma)} \left(|[k_\beta]| \frac{x_\gamma y_\beta}{1 + x_\gamma y_\beta} \right) + (|[k_\gamma]| - 1) \frac{x_\gamma y_\gamma}{1 + x_\gamma y_\gamma} \tag{6.46}$$

and analogously for the in-degree

$$k_\gamma^{in} = \sum_{\beta(\neq\gamma)} \left(|[k_\beta]| \frac{y_\gamma x_\beta}{1 + y_\gamma x_\beta} \right) + (|[k_\gamma]| - 1) \frac{y_\gamma x_\gamma}{1 + y_\gamma x_\gamma}. \tag{6.47}$$

6.4 Iterative resolution of the DBCM

Being able to reduce the dimensionality of a problem is a huge step towards a faster resolution of it. Let us now combine the reduction of dimensionality with an alternative method to solve coupled, non-linear systems of equations.

As we said in section 6.1, solving an Exponential Random Graph Model means solving an optimization problem. To this aim, many algorithms exist (e.g. Newton's method, the 4-th order Runge-Kutta one): still they all tend to be computationally expansive. An alternative approach is represented by the method proposed in Dianati (2016) where it is applied to bipartite networks; here, we are going to discuss a slightly modified version of it, designed for solving the DBCM.

The idea presented in Dianati (2016) is indeed simple: rewriting the first-order equations (6.28) and (6.29) in an *iterative* fashion, i.e. as to find the fixed-point of the system of equations

$$x_i = \frac{k_i^{out}(\mathbf{A}^*)}{\sum_{j(\neq i)=1}^{N} \left(\frac{y_j}{1 + x_i y_j} \right)}, \quad \forall\, i \tag{6.48}$$

$$y_i = \frac{k_i^{in}(\mathbf{A}^*)}{\sum_{j(\neq i)=1}^{N} \left(\frac{x_j}{1 + y_i x_j} \right)}, \quad \forall\, i \tag{6.49}$$

defining the DBCM. To this aim, let's define the maps

$$x_i = \psi(\underline{x}, \underline{y}) \quad \text{and} \quad y_j = \chi_j(\underline{x}, \underline{y}), \quad \forall\, i, j \tag{6.50}$$

that represent the equations (6.48) and (6.49). The solution to the system is, then, the fixed-point of the iterative map defined by ψ_i and χ_j, $\forall\, i, j$. Starting from an initial point one construct the iteration as:

$$x_i^{k+1} = \psi_i(\underline{x}^k, \underline{y}^k), \quad \forall\, i \tag{6.51}$$
$$y_j^{k+1} = \chi_j(\underline{x}^k, \underline{y}^k), \quad \forall\, j \tag{6.52}$$

where k indicates the k-th iterative step.

For what concerns the starting point, contributions in the literature suggest to employ $x_i = \frac{k_i^{out}(\mathbf{A}^*)}{\sqrt{L}}$ and $y_i = \frac{k_i^{in}(\mathbf{A}^*)}{\sqrt{L}}$, $\forall\, i$; other initial conditions can be chosen by drawing the starting point from a uniform distribution between 0 and 1. The method outlined here is very simple and can be easily adapted to solve the reduced system of equations previously introduced: we only need to recall the first-order conditions for the reduced DBCM in equations (6.46) and (6.47) and, then, define the *ad hoc* map

$$x_\alpha = \psi_\alpha^{red}(\underline{x}, \underline{y}) = \frac{k_\alpha^{out}(\mathbf{A}^*)}{\sum_{\alpha(\neq\beta)}(|[k_\beta]| - \mathbb{1}_{\alpha=\beta})\frac{y_\beta}{1+x_\alpha y_\beta}}, \quad \forall\, \alpha \tag{6.53}$$

$$y_\gamma = \chi_\gamma^{red}(\underline{x}, \underline{y}) = \frac{k_\gamma^{in}(\mathbf{A}^*)}{\sum_{\beta(\neq\gamma)}(|[k_\beta]| - \mathbb{1}_{\gamma=\beta})\frac{x_\beta}{1+y_\gamma x_\beta}}, \quad \forall\, \gamma \tag{6.54}$$

whose resolution translates into finding its fixed point, i.e.

$$(\underline{x}, \underline{y}) = (\psi^{red}, \chi^{red})(\underline{x}, \underline{y}) \tag{6.55}$$

6.5 Solving the DBCM on Bitcoin: examples

Let us now explicitly solve the DBCM on our weekly BUNs, constructed from the data on Bitcoin transactions, across a period of time going from

Data: $\underline{z}_0 = (\underline{x}_0, \underline{y}_0)$ initialize the map to solve the reduced DBCM.
Result: $\underline{z} = (\underline{x}, \underline{y})$ solution of the DBCM problem.
while $\|\underline{z} - \underline{z}^{old}\| > \epsilon$ **do**
  $\underline{z} = \phi(\underline{z}|\mathbf{A}^*)$
  $\underline{z}^{old} = \underline{z}$
end

Algorithm 1: Pseudo-code for solving the reduced version of the DBCM via the iterative map.

2009 to mid-2018. As explained in chapter 1, nodes are users and links are flows of bitcoins (aggreagted across weeks) between them.

To overcome the limitations encountered when implementing full numerical approaches, we consider the reduced, numerical ones. This greatly reduces the dimensionality of the problem: as shown in figure 31, the size of the reduced system is three orders of magnitude smaller than that of the full one: in fact, the number of nodes decreases from $4 \cdot 10^6$ to $6 \cdot 10^3$ (at its peak). To further speed up the resolution of the problem, we have implemented the iterative algorithm - see equations (6.48) and (6.49) - whose pseudo-code is displayed as algorithm 1. Our criterion for defining convergence has been quantified by setting the tolerated difference between subsequent iterations at 10^{-6}.

Figure 32 reports some analytics about the performance of our algorithm. Plotting the number of steps required by our fixed-point algorithm to reach convergence, as a function of the networks size, reveals, with no surprise, that it rises with N; however, the total amount of time required to reach convergence is of (the order of) hundreds of seconds, for configurations with more than one million of nodes.

Let us now inspect the goodness of our solution, by plotting the error made by our algorithm as a function of the number of nodes. The error is defined as the maximum of the (absolute value of the) difference between the empirical and the expected degrees, i.e.

$$\Delta = \max \left\{ |k_i^{out,*} - \langle k_i^{out} \rangle|, |k_i^{in,*} - \langle k_i^{in} \rangle| \right\}_{i=1}^{N}; \tag{6.56}$$

as shown in figure 32, the error is $\Delta \gtrsim 10^{-2}$: in other words, the largest

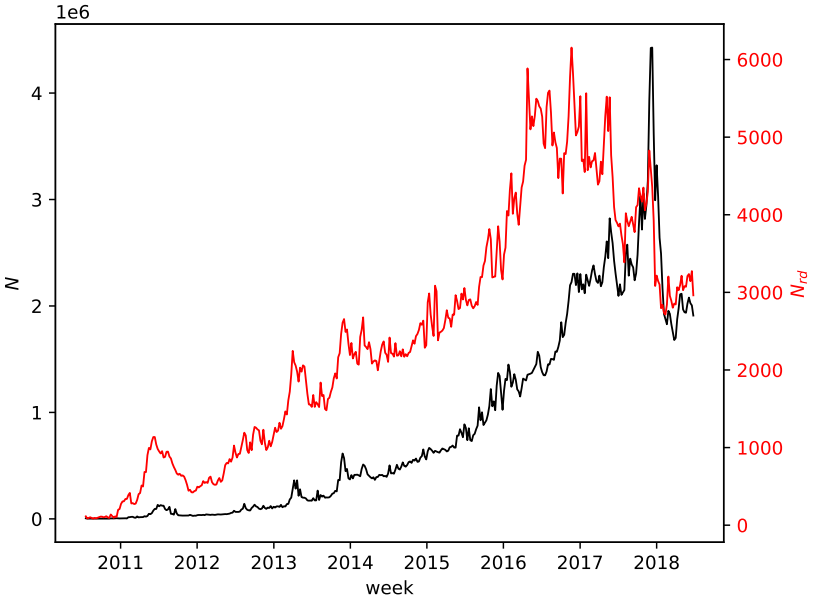

Figure 31: Evolution of the dimension of the full DBCM (in black) and of the dimension of the reduced DBCM (in red), calculated for the Bitcoin User Networks: the latter one is three orders of magnitude smaller than the former one.

error we make is, in absolute terms, of the order of 0.01. Remarkably, larger configurations characterized by a smaller error (with convergence requiring some more time) exist.

6.6 Sampling the DBCM ensemble

Let us now describe how to sample the DBCM ensemble, induced by the degrees characterizing the empirical, binary, directed graph $\mathbf{A}^*$: solving the maximum entropy problem described in section 6.3 leads us to find the vector of solution parameters $\underline{\theta}$ by means of which we can define the numerical value of the link probabilities

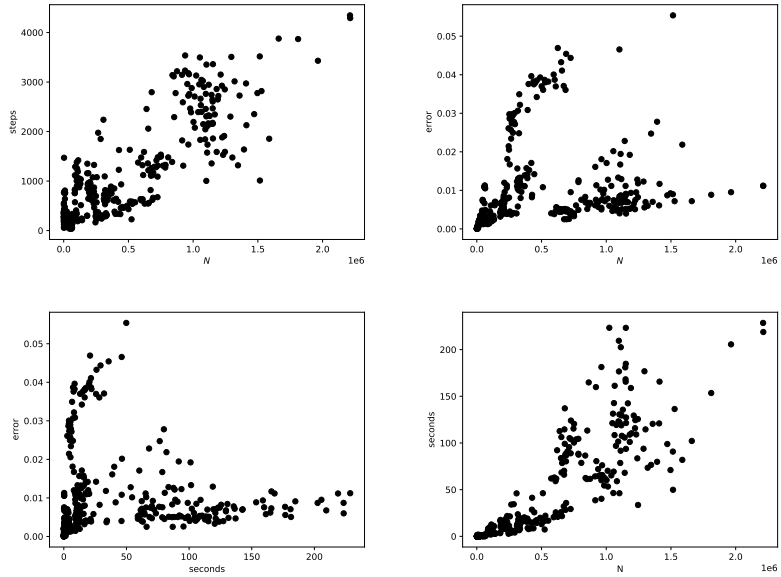

Figure 32: Resolution of the reduced version of the DBCM for the BUNs at the weekly time scale: plotting the time (in seconds) and the error versus the total number of nodes reveals that the total amount of time required to reach convergence is of (the order of) hundreds of seconds, for configurations with more than one million of nodes. However, a non-linear relationship between solving time and error exists: on some configurations the DBCM has been solved in $\simeq 5$ seconds, with an error of $\simeq 0.05$; on others, its resolution has required $\simeq 200$ seconds (i.e. 4 minutes), with an error of $\simeq 0.01$.

$$p_{ij}^{\text{DBCM}} = \frac{x_i y_j}{1 + x_i y_j}, \quad \forall\, i \neq j; \tag{6.57}$$

since $a_{ij} \sim \text{Ber}[p_{ij}^{\text{DBCM}}]$, i.e. the presence of a link from node i to node j is described by a Bernoulli probability distribution, generating an entire graph amounts at repeating the same process for each of the $N(N-1)$ entries of its adjacency matrix[1]. The pseudo-code for the process is the

[1]Diagonal entries are excluded since self-loops are usually ignored.

Data: $\underline{\theta} = (\underline{\theta}^{out}, \underline{\theta}^{in})$ solution of the DBCM problem.
Result: adjacency matrix drawn from the DBCM ensemble induced by $\underline{\theta}$.

```
for i = 1...N do
    for j = 1...N do
        if i ≠ j then
            draw r from a uniform distribution defined on [0, 1]; if
             r ≤ p_ij^DBCM then
                a_ij = 1;
            else
                a_ij = 0;
            end
        end
    end
end
```

Algorithm 2: Pseudo-code for sampling one adjacency matrix from the DBCM ensemble.

one shown by Algorithm 1.

While the algorithm described above is fairly simple, it quickly becomes computationally expensive since the number of entries to be checked is $O(N^2)$ - to give an idea, for systems of the size of Bitcoin in 2018, i.e. of millions of nodes, this would amount at doing 10^{12} checks; even more so, this is the recipe to generate just one graph - in order to obtain a representative sample of the DBCM ensemble, one may need to generate thousands of graphs. Hence, while very simple, the sampling procedure above is not appealing to provide ensemble estimates of network properties for (very) large systems.

6.7 The Delta method

Luckily, ERGMs allow one to carry out analytical calculations of the expectation value as well as of the standard deviation of a wide range of topological properties. While formulas require some approximations, they often represent the most viable alternative to get the searched values.

Given a topological property X we can define the average of X over any ensemble as

$$\langle X \rangle = \sum_{\mathbf{G}} X(\mathbf{G}) P(\mathbf{G}|\underline{\theta}) \tag{6.58}$$

where $\langle X \rangle$ is the expected value of X on the chosen ensemble, i.e. induced by the constraints defining it (see also Squartini et al. (2011)). Comparing $X(\mathbf{G}^*)$ (i.e. the empirical value of the chosen property) with the ERGM average $\langle X \rangle$ provides information about whether the constraints used to generate the ERGM are able to explain the empirical value $X(\mathbf{G}^*)$ or additional information is needed. Naturally, the value of the topological properties used as constraints to generate the ensemble corresponding to the chosen ERGM are reproduced once the maximum of the likelihood principle is employed to estimate $\underline{\theta}$.

Let us focus on binary networks (hence, changing the notation from $\mathbf{G}$ to $\mathbf{A}$). The simplest quantity for which we can compute the ensemble average is a_{ij}:

$$\langle a_{ij} \rangle = \sum_{\mathbf{A}} a_{ij} P(\mathbf{A}|\underline{\theta}); \tag{6.59}$$

when replacing the generic $P(\mathbf{A}|\underline{\theta})$ with the probability distributions of the UBCM and the DBCM, the result above translates into

$$\langle a_{ij} \rangle_{\text{UBCM}} = \frac{x_i x_j}{1 + x_i x_j} \equiv p_{ij}^{\text{UBCM}}, \tag{6.60}$$

$$\langle a_{ij} \rangle_{\text{DBCM}} = \frac{x_i y_j}{1 + x_i y_j} \equiv p_{ij}^{\text{DBCM}} \tag{6.61}$$

as we saw in equations (6.26) and (6.15). This simple relationship allows us to calculate the expected value of a number of topological quantities exactly. An example is provided by the dyadic motifs, defined in equations (4.27), (4.28) and (4.29): upon considering that, under the UBCM and the DBCM, dyads are independent, one finds that

$$\langle L^{\leftrightarrow} \rangle = \sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} p_{ij} p_{ji}, \tag{6.62}$$

$$\langle L^{\rightarrow} \rangle = \sum_{i=1}^{N} \sum_{j(>i)=1}^{N} [p_{ij}(1-p_{ji}) + p_{ji}(1-p_{ij})], \tag{6.63}$$

$$\langle L^{\not\leftrightarrow} \rangle = \sum_{i=1}^{N} \sum_{j(\neq i)=1}^{N} (1-p_{ij})(1-p_{ji}). \tag{6.64}$$

In general, however, solving the ERGM problem doesn't ensure that equation (6.58) can be calculated since this depends both on the model employed and on the topological property one aims at computing. In this cases, one needs to proceed by approximations: if we define the gradient matrix of a topological property $X(\mathbf{A})$ as $\nabla_{ij} X(\mathbf{A}) = \frac{\partial X(\mathbf{A})}{\partial a_{ij}}$ and denote by $\langle \mathbf{A} \rangle \equiv \mathbf{P}$ the matrix whose generic entry reads $\langle a_{ij} \rangle \equiv p_{ij}$, then the first-order Taylor expansion of X around $\langle \mathbf{A} \rangle$ reads

$$\begin{aligned} X(\mathbf{A}) &= X(\langle \mathbf{A} \rangle) + \sum_{i,j} (a_{ij} - \langle a_{ij} \rangle) \frac{\partial X(\mathbf{A}))}{\partial a_{ij}} \Big|_{\mathbf{A}=\langle \mathbf{A} \rangle} + \dots \\ &= X(\langle \mathbf{A} \rangle) + (\mathbf{A} - \langle \mathbf{A} \rangle) \cdot \nabla X(\langle \mathbf{A} \rangle) + \dots \\ &= X(\mathbf{P}) + (\mathbf{A} - \mathbf{P}) \cdot \nabla X(\mathbf{P}) + \dots \end{aligned} \tag{6.65}$$

where $A \cdot B = \sum_{\substack{i,j \\ i \neq j}} a_{ij} b_{ij}$ is a compact notation for the matrix scalar product. By taking the expected value of both sides we obtain

$$\langle X \rangle \simeq X(\mathbf{P}) \tag{6.66}$$

since the terms beyond the first-order ones disappear. This is the first step of the so called *Delta method*. As we will show, the approximation above is often a good one. To be noticed that in case a linear quantity is considered (e.g. one of the constraints defining the models we have considered here), substituting a_{ij} with p_{ij} provides the exact expectation of the quantity under consideration.

Interestingly, dyads provide another example of quantities whose expectation can be exactly recovered by replacing a_{ij} with p_{ij}; approximate expressions, instead, are the ones defining the expected value of the directed versions of the ANND (e.g. $\langle k_i^{out,out} \rangle_{\text{DBCM}}$). The formulas above also clarify why the computational time is reduced by analytic calculations: instead of generating thousands of graphs and averaging on the dyadic variables of each, we just need to evaluate a single formula. However, the computational time of the formula itself cannot be reduced to a much lesser extent; let us imagine a quantity defined by a summation over all node pairs: its expected value will require the same number of sums as well.

The second step of the so called *Delta method* concerns the computation of the standard deviation of a network topological property. First, let's write the variance of a topological property X as:

$$\sigma^2[X] = \langle (X - \langle X \rangle)^2 \rangle = \langle X^2 \rangle - \langle X \rangle^2 \tag{6.67}$$

an expression that depends on $\underline{\theta}$ as $\langle X \rangle$ does. By substituting the linear approximation introduced in equation (6.65) one finds

$$\sigma^2[X] = \sum_{i,j} \sum_{t,s} \sigma[a_{ij}, a_{ts}] \left(\frac{\partial X(\mathbf{A})}{\partial a_{ij}} \cdot \frac{\partial X(\mathbf{A}}{\partial a_{ts}} \right) \Bigg|_{\mathbf{A}=\langle \mathbf{A} \rangle} + \dots \tag{6.68}$$

where the covariance of a_{ij} and a_{ts} is

$$\begin{aligned} \sigma[a_{ij}, a_{ts}] &= \langle (a_{ij} - \langle a_{ij} \rangle)(a_{ts} - \langle a_{ts} \rangle) \rangle & (6.69) \\ &= \langle a_{ij} a_{ts} \rangle - \langle a_{ij} \rangle \langle a_{ts} \rangle & (6.70) \end{aligned}$$

(having defined $\langle a_{ij} a_{ts} \rangle = \sum_{\mathbf{A}} a_{ij} a_{ts} P(\mathbf{A}|\underline{\theta})$); the standard deviation of the topological property X, then, reads

$$\sigma[X] = \sqrt{\sum_{i,j} \sum_{t,s} \sigma[a_{ij}, a_{ts}] \left(\frac{\partial X(\mathbf{A})}{\partial a_{ij}} \cdot \frac{\partial X(\mathbf{A}}{\partial a_{ts}} \right) \Bigg|_{\mathbf{A}=\langle \mathbf{A} \rangle} + \dots} \tag{6.71}$$

Equation (6.71) completes the recipe for calculating the average and the standard deviation of any network quantity of interest - in case it were not possible to obtain an exact, analytical result. Examples are provided by the constraints (more in general, by linear quantities) or by quantities as the aforementioned dyadic motifs.

Once the average and the standard deviation of a quantity of interest have been obtained, one can asses whether the empirical value $X(\mathbf{A}^*)$, observed on the original network $\mathbf{A}^*$, is consistent with the randomized value $\langle X \rangle$: to this aim, one can compute the inequality

$$|X(\mathbf{A}^*) - \langle X \rangle| \tag{6.72}$$

and determine how many standard deviations $X(\mathbf{A}^*)$ differs from $\langle X \rangle$. Alternatively, one can compute the z-score

$$z[X] = \frac{X(\mathbf{A}^*) - \langle X \rangle}{\sigma[X]} \tag{6.73}$$

whose interpretation has been provided in chapter 3.

Chapter 7

Epilogue

Todo lo que empieza como comedia
acaba como ejercicio criptográfico.

Roberto Bolaño

In this thesis we studied several Bitcoin Transaction Networks. We offered a different way to look at cryptocurrencies, alternative and complementary to economics and computer science, adopting a large variety of methodologies from different fields of knowledge in order to offer a comprehensive, yet deep, analysis.

One of our main findings concerns the growing centralization of Bitcoin Transaction Networks. As we observed, Bitcoin Transaction Networks become more centralized as time goes by: as an example, we notice the emergence of hubs, i.e. hyper-connected nodes which are in between the majority of transactions. This results contributes to the ongoing debate about cryptocurrencies and (de)centralization.

Cryptocurrencies are related to decentralization in a two-fold way: *internally* and *externally*. From the inside, Bitcoin functioning as a currency rests upon its decentralized nature: peers store redundant copies of the blockchain, no central institution exists but many individuals, mining is a (decentralized) process substituting a (central) validator. This is reflected on the Bitcoin goal, i.e. that of defining a decentralised finance

via a decentralized currency which is not controlled by anyone, therefore providing true safety against third parties (often identified as big corps or governments) interference. The (sometimes vague) concept of decentralization lies at the very core of the entire Bitcoin narrative.

On the other hand, Bitcoin increasing centralization was observed under many respects. The most relevant example is represented by the increasing concentration of computational power into the hands of mining pools: as we depicted in Chapter 2, Bitcoin relies on the ever-increasing competition among miners to ensure blockchain immutability. The emergence of an actor able to control over 51% of the total computational power would imply that such a miner could take over the blockchain writing process - therefore, Bitcoin itself. While this is not the case (yet), recently, miners gathered together into the so-called mining pools, to allow 'small' miners to remain competitive. Therefore, a small number of huge actors (i.e. the mining pools) indeed control the writing process of Bitcoin, at the moment. This is not only a 'philosophical' threat to Bitcoin ideas but represents a physical threat for the existence of the Bitcoin ecosystem.

Why do Bitcoin Transaction Networks tend to be so centralized? Is Bitcoin robustness (anonymity, resilience to external attacks, etc.) somehow reduced because of it? Is centralization a common feature of transaction networks? An infamous dilemma is whether Bitcoin is understood as a mean to exchange value or just a vector of speculation: the observed centralized structures would suggest the correct answer to be the latter. In a transaction network where people are actually using tokens as a currency one would expect nodes interacting with each other[1]. To provide an intuition of this, think of your everyday-life transactions: while, if we were to track the Euro transactions network in a country like Italy, banks, big retailers franchising, etc. would emerge as super-connected nodes, we would also observe a large number of nodes with connectivity larger than the average user, representing family businesses, restaurants, etc. The share of 'small nodes transactions' seems to be absent in the Bitcoin

[1]Of course we would expect large hubs to emerge but also nodes with medium-to-small degree to be (very) frequent.

ecosystem, corroborating the thesis that users join Bitcoin only for speculation purposes, i.e. to invest and store value instead of using it as a real currency.

If this is a common feature of all cryptocurrencies is a question for future work - something we are already working on. At the time this thesis is being written, I am a member of the UZH Blockchain Observatory Centre and currently working to extend the analysis presented in this thesis to other blockchain-based cryptocurrencies. Our goal, now, is that of looking for similarities between cryptocurrencies transaction networks, in order to spot the presence of general laws for the structural growth of these entities.

Another interesting topic I'm currently moving my attention towards is consensus in Distributed Ledger Technology (the blockchain is a distributed ledger technology): generalizing the Bitcoin consensus mechanism is an exciting and innovative research field lying at the basis of future cryptocurrencies.

On the more theoretical side, research on statistical tools does not stop with the current work as well: at the moment, I am the maintainer, together with my co-authors, of the Python 3 module 'NEMtropy' (the acronym standing for Network Entropy Maximization: a toolbox running on Python[2]) that implements the null models employed for the present analysis - together with many others. We hope to be able to keep improving the code for a long time - maybe providing other PhD students with a platform where their own codes can be uploaded.

[2] `https://pypi.org/project/NEMtropy/`

Bibliography

Aldecoa, R and I Marin (2013a). "Exploring the limits of community detection strategies in complex networks". In: *Scientific Reports* 3.2216.

— (2013b). "Surprise maximization reveals the community structure of complex networks". In: *Scientific Reports* 3.1060.

Androulaki, Elli et al. (2013). "Evaluating user privacy in bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer, pp. 34–51.

Antonopoulos, Andreas M (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies*. O'Reilly Media, Inc.

Aspembitova, A et al. (2019). "Fitness preferential attachment as a driving mechanism in bitcoin transaction network". In: *PLoS ONE* 14.8, e0219346. DOI: 10.1371/journal.pone.0219346.

Bauke, Heiko (2007). "Parameter estimation for power-law distributions by maximum likelihood methods". In: *The European Physical Journal B* 58.2, pp. 167–173.

Bergstra, Jan A and Peter Weijland (2014). "Bitcoin: a money-like informational commodity". In: *arXiv preprint arXiv:1402.4778*.

Bouoiyour, Jamal and Refk Selmi (2015). "What does Bitcoin look like?" In: *Annals of Economics and Finance* 16.2, pp. 449–492.

Bovet, Alexandre et al. (2019). *The evolving liaisons between the transaction networks of Bitcoin and its price dynamics*. arXiv: 1907.03577 [q-fin.GN].

Brands, Stefan (1993). "Untraceable off-line cash in wallet with observers". In: *Annual international cryptology conference*. Springer, pp. 302–318.

Chaum, David (1983). "Blind signatures for untraceable payments". In: *Advances in cryptology*. Springer, pp. 199–203.

Dai, Wei (1998). "B-money proposal". In: *White Paper*.

Decker, Christian and Roger Wattenhofer (2014). "Bitcoin Transaction Malleability and MtGox". In: *Computer Security - ESORICS 2014*. Ed. by Mirosław Kutyłowski and Jaideep Vaidya. Cham: Springer International Publishing, pp. 313–326. ISBN: 978-3-319-11212-1.

Di Francesco Maesa, D, A Marino, and L Ricci (2016a). "An analysis of the Bitcoin users graph: inferring unusual behaviours". In: *International Workshop on Complex Networks and their Applications*. Springer, Cham, pp. 749–760.

— (2016b). "Uncovering the Bitcoin blockchain: an analysis of the full users graph". In: *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, pp. 537–546.

— (2017). "Detecting artificial behaviours in the Bitcoin users graph". In: *Online Soc. Netw. Media* 3, pp. 63–74.

— (2018a). "Data-driven analysis of Bitcoin properties: exploiting the users graph". In: *Int. J. Data Sci. Anal.* 6.1, pp. 63–80.

— (2018b). "The graph structure of Bitcoin". In: *International Conference on Complex Networks and Their Applications*. Springer, Cham, pp. 547–558.

— (2019). "The bow tie structure of the Bitcoin users graph". In: *App. Netw. Science* 4.56.

Dianati, Navid (2016). "A maximum entropy approach to separating noise from signal in bimodal affiliation networks". In: *arXiv preprint arXiv:1607.01735*.

Dwivedi, A D et al. (2019). "A decentralized privacy-preserving healthcare blockchain for IoT". In: *Sensors* 19.2.

Fantazzini, Dean, Sergey Ivliev, and Vera Sukhanovskaya (2017). "Everything you always wanted to know about bitcoin modelling but were afraid to ask. Part 2". In: *Applied Econometrics* 45, pp. 5–28.

Fergal, R and M Harrigan (2013). "An analysis of anonymity in the bitcoin system". In: *Security and privacy in social networks*. Springer, New York, pp. 197–223.

Garcia, David et al. (2014). "The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy". In: *Journal of the Royal Society Interface* 11.99, p. 20140623.

Garlaschelli, Diego and Maria I Loffredo (2008). "Maximum likelihood: extracting unbiased information from complex networks". In: *Physical Review E* 78.1, p. 015101.

Glatterfelder, James B. (Sept. 2019). "The Bow-Tie centrality: a novel measure for directed and weighted networks with an intrinsic node property". In: *Advances in Complex Systems* 22.06, p. 1950018. ISSN: 1793-

6802. DOI: 10.1142/s0219525919500188. URL: http://dx.doi.org/10.1142/S0219525919500188.

Gradojevic, Nikola (2014). "Foreign exchange customers and dealers: Who's driving whom?" In: *Finance Research Letters* 11.3, pp. 213–218.

Granger, Clive WJ (1969). "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: journal of the Econometric Society*, pp. 424–438.

Harrigan, M and C Fretter (2016). "The unreasonable effectiveness of address clustering". In: *Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, pp. 368–373.

Hong, Yongmiao, Yanhui Liu, and Shouyang Wang (2009). "Granger causality in risk and detection of extreme risk spillover between financial markets". In: *Journal of Econometrics* 150.2, pp. 271–287.

Jakobsson, Markus and Ari Juels (1999). "Proofs of Work and Bread Pudding Protocols(Extended Abstract)". In: *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium*. Ed. by Bart Preneel. Boston, MA: Springer US, pp. 258–272. DOI: 10.1007/978-0-387-35568-9_18. URL: https://doi.org/10.1007/978-0-387-35568-9_18.

Javarone, Marco Alberto and Craig Steven Wright (2018). *From Bitcoin to Bitcoin Cash: a network analysis*. Tech. rep. arXiv: 1804.02350.

Jeude, J van Lidth de, Guido Caldarelli, and Tiziano Squartini (2019). "Detecting core-periphery structures by surprise". In: *EPL (Europhysics Letters)* 125.6, p. 68001.

Kondor, D et al. (2014). "Do the rich get richer? An empirical analysis of the bitcoin transaction network". In: *PLoS ONE* 9.2, p. 86197.

Konig, M D, Claudio Tessone, and Y Zenou (2010). "From assortative to dissortative networks: the role of capacity constraints". In: *Advances in Complex Systems* 13.4, p. 483.

— (2014). "Nestedness in networks: a theoretical model and some applications". In: *CEPR Discussion Paper no. 7521* 9, pp. 695–752.

Latora, Vito, Vincenzo Nicosia, and Giovanni Russo (2017). *Complex networks: principles, methods and applications*. Cambridge University Press.

Lidth de Jeude, Jeroen van et al. (2019). "Reconstructing mesoscale network structures". In: *Complexity* 2019, p. 5120581. DOI: 10.1155/2019/5120581.

Lin, Jian-Hong et al. (Aug. 2020). "Lightning network: a second path towards centralisation of the Bitcoin economy". In: *New Journal of Physics* 22.8, p. 083022. DOI: 10.1088/1367-2630/aba062.

Lischke, Matthias and Benjamin Fabian (2016). "Analyzing the Bitcoin network: the first four years". In: *Future Internet* 8.1. DOI: 10.3390/fi8010007.

Meiklejohn, S et al. (2013). "A fistful of bitcoins: characterizing payments among men with no names". In: *Proceedings of the 2013 conference on Internet measurement conference*, pp. 127–140.

Metcalfe, Bob (2013). "Metcalfe's law after 40 years of ethernet". In: *Computer* 46.12, pp. 26–31.

Morgan, James (1962). "The anatomy of income distribution". In: *The review of economics and statistics*, pp. 270–283.

Nakamoto, Satoshi (2019). *Bitcoin: A peer-to-peer electronic cash system*. Tech. rep. Manubot.

Newman, Mark (2003). "Mixing patterns in networks". In: *Physical review E* 67.2, p. 026126.

— (2018). *Networks*. Oxford university press.

Ober, M, S Katzenbeisser, and K Hamacher (2013). "Structure and anonymity of the Bitcoin transaction graph". In: *Future Internet* 5.2, pp. 237–250.

Okamoto, Tatsuaki and Kazuo Ohta (1991). "Universal electronic cash". In: *Annual international cryptology conference*. Springer, pp. 324–337.

Park, Juyong and Mark EJ Newman (2004). "Statistical mechanics of networks". In: *Physical Review E* 70.6, p. 066117.

Poon, Joseph and Thaddeus Dryja (2016). *The bitcoin lightning network: Scalable off-chain instant payments*.

Popuri, M K and Gunes M H (2016). "Empirical analysis of crypto currencies". In: *Complex Networks* VII, pp. 281–292.

Reid, Fergal and Martin Harrigan (2013). "An analysis of anonymity in the bitcoin system". In: *Security and privacy in social networks*. Springer, pp. 197–223.

Restocchi, Valerio, Frank McGroarty, and Enrico Gerding (2019). "The stylized facts of prediction markets: analysis of price changes". In: *Physica A: Statistical Mechanics and its Applications* 515, pp. 159–170.

Rohrer, E, J Malliaris, and F Tschorsch (2019). "Discharged payment channels: quantifying the Lightning Network resilience to topology-based attacks". In: *arXiv:1904.10253*.

Rombach, M Puck et al. (2014). "Core-periphery structure in networks". In: *SIAM Journal on Applied mathematics* 74.1, pp. 167–190.

Ron, Dorit and Adi Shamir (2013). "Quantitative analysis of the full bitcoin transaction graph". In: *International Conference on Financial Cryptography and Data Security*. Springer, pp. 6–24.

Singh, Sunil Kumar and Sumit Kumar (2021). *Blockchain technology: introduction, integration and security issues with IoT*. Tech. rep. arXiv: 2101.10921.

Sornette, Didier and Peter Cauwels (2014). "Financial bubbles: mechanisms and diagnostics". In: *Swiss Finance Institute Research Paper* 14-28.

Squartini, Tiziano and Diego Garlaschelli (2011). "Analytical maximum-likelihood method to detect patterns in real networks". In: *New Journal of Physics* 13.8, p. 083001.

Stadler, Markus, Jean-Marc Piveteau, and Jan Camenisch (1995). "Fair blind signatures". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, pp. 209–219.

Sultan, K, U Ruhi, and R Lakhani (2018). *Conceptualizing blockchains: characteristics and applications*. Tech. rep. arXiv: 1806.03693.

Tasca, Paolo, Adam Hayes, and Shaowen Liu (2018). "The evolution of the bitcoin economy: extracting and analyzing the network of payment relationships". In: *J. Risk Financ.* 19.2, pp. 94–126.

Vallarano, Nicolò, Claudio J. Tessone, and Tiziano Squartini (2020). "Bitcoin Transaction Networks: an overview of recent results". In: *Frontiers in Physics* 8, p. 286.

— (2021a). "Exploring the Bitcoin mesoscale structure". In: *in preparation*.

Vallarano, Nicolò et al. (2021b). "Fast and scalable likelihood maximization for Exponential Random Graph Models". In: *arXiv:2101.12625v1*.

Van Alstyne, Marshall (2014). "Why Bitcoin has value". In: *Communications of the ACM* 57.5, pp. 30–32.

Wallace, David AR (2012). *Groups, rings and fields*. Springer Science & Business Media.

Wheatley, Spencer et al. (2018). "Are Bitcoin bubbles predictable? Combining a generalized Metcalfe's Law and the LPPLS model". In: *Combining a Generalized Metcalfe's Law and the LPPLS Model (March 15, 2018). Swiss Finance Institute Research Paper* 18-22.

Wu, Ke, Wheatley, Spencer, and Sornette, Didier (2018). "Classification of cryptocurrency coins and tokens by the dynamics of their market capitalizations". In: *Royal Society open science* 5.9, p. 180381.

Yermack, David (2015). “Is Bitcoin a real currency? An economic appraisal”. In: *Handbook of digital currency*. Elsevier, pp. 31–43.

Glossary

address A Bitcoin address is an alphanumerical string with a unction very similar to a physical address or an email address. It is the only information needed to send bitcoins to a user. It is a good practice that of using Bitcoin addresses only once. 112

Bitcoin The whole Bitcoin ecosystem, i.e. the public ledger, the transaction network, the digital token, the transaction protocol, etc. ix, xiii–xv, 5, 7–24, 48, 54, 61, 65

bitcoin The digital token exchanged via the Bitcoin protocol, originally introduced in Nakamoto (2019). 6–8, 10, 11, 17, 24, 26, 56

Bitcoin Transaction Networks The networks of monetary transactions in bitcoins. In the present work, we distinguish between the Bitcoin Address Network, where the nodes are Bitcoin addresses and the links are the transactions among them, and the Bitcoin User Network, where the nodes are users (i.e. clusters of addresses) and the links are the aggregated transactions among the addresses that define the users. To have a proper definition of a 'network of transactions' one needs to set a time window to select the transactions themselves. 24, 27, 40, 62

block Atomic element constituting the blockchain, that contains all verified transactions. A block is made of an *header* and a *body*. All transactions are written in the body, while the *header* contains meta-

information such as the timestamp, the block number, the previous block hash, etc. 8, 11, 12, 19, 20

blockchain The public ledger on which the Bitcoin transactions are recorded. The blockchain can be thought as a list (or a *chain*) of single blocks (which are the elements containing the information on the actual transactions) where they are 'attached' one by one. On average, a new block appears every ten minutes. ix, 5, 9, 10, 12–14, 18–20, 102

DBCM The Directed Binary Configuration Model belongs to the class of Exponential Random Graph Models. Given the in- and out-degree sequences of a network, the model allows one to generate an ensemble of graphs which, on average, have the same in- and out-degrees of the empirical one. xii, xvii, 44–46, 61–64, 72, 85, 86, 89, 91, 92, 94–97

degree A local network measure. For undirected networks, the degree of a node counts the number of (other) nodes connected to it. For directed networks, the notions of in- and out-degrees can be introduced, respectively counting the number of neighbors pointing to it and it points to. 23, 37, 39–41, 62, 67, 71, 73, 75

Distributed Ledger Technology A distributed ledger database spread across several devices on a peer-to-peer network. A consensus algorithm is required to assure nodes information is coherent. Pure DLTs requires no central authority to regulate themselves. The blockchain is one example of distributed ledger design.. 103

ensemble In statistical physics, an ensemble (also *statistical ensemble*) is a collection of a large number of copies (sometimes, infinitely many) of a system, each of which represents a possible state in which a real system can be found in. An ensemble is also the support of a probability distribution for the states of the system. xiv, xv, 62–65, 72, 78, 79, 82, 85, 96, 97, 111, 112

input A transaction input is the set of addresses 'providing' bitcoins to the transaction. 8, 10, 18

mining pool A mining pool is a group of miners who pool together their computational resources in order to mine and split together eventual rewards.. 102

output A transaction output is the set of addresses receiving bitcoins 'as a consequence' of the transaction. 8–10, 14, 18

proof-of-work The proof-of-work (PoW) is a form of cryptographic zero-knowledge proof in which one party (the prover) proves to others (the ones who verify) that a certain amount of computational effort has been spent, for some purpose, in a given amount of time (see Jakobsson et al. (1999)). In the Bitcoin protocol, the proof-of-work is used by miners to prove that they spent computational work to verify blocks (i.e. transactions). The amount of computational work required to verify a new block is automatically adjusted to require, on average, ten minutes. 11

Shannon entropy In information theory, the Shannon Entropy (also known as *information entropy*) of a random variable is the average level of 'information', 'surprise' or 'uncertainty' inherent to the variable possible outcomes. 78–80

UBCM The Undirected Binary Configuration Model belongs to the class of Exponential Random Graph Models. Given the degree sequence of a network, the model allows one to generate an ensemble of graphs which, on average, have a degree sequence equal to the empirical one. 72–74, 80–82, 85, 89, 90, 97