Computational Information Density and Entropy of the Bitcoin blockchain

C.T. Nesenberend

July 31, 2020

Abstract

In econophysics, statistical-physics techniques are used to model economical systems. In this thesis, we investigate the entropy and the Computational Information Density (CID) of the Bitcoin blockchain. The CID is defined as the compression ratio of some particular algorithm when applied to the raw data of the state of the system. It is related to entropy as both CID and entropy are measures of information.

We find a strong correspondence between the CID and entropy for the Bitcoin blockchain, where features are similar, but without one being a clear function of the other. This can be explained by intercorrelations between one agent and the next, which the entropy does not count. We also calculate some correlations to see if the CID and the entropy have some predictive power for the price, and we find a small correlation, but very small in comparison to the predictive power of the price itself.

These results the power of the CID-entropy correspondence and how the Bitcoin blockchain may be used as a useful large-scale toy model for econophysics. We anticipate that these results can be used for a further look into the CID-entropy relation, as the similarities are visible but there is no exact correspondence. Besides this, these results can form a basis for a further look into the predictive power of the CID or the entropy for the price.

Contents

Chapter 1

Introduction

Statistical physics describes systems with many particles which are behaving similarly. However, these particles need not be elemental particles, they can be composite particles. With the right interaction terms these statistical models of composite particles are still very accurate. Now, why not take this to the extreme, and consider huge "particles", humans (or other agents in economical systems)? This way we can try to describe our economical systems in a physical way. This is the idea behind econophysics, in which systems of many similarly acting agents model the economy, and statistical-physics techniques are used to infer properties of the system [1, 2].

What we are studying in this thesis in particular are the concepts of entropy and Computational Information Density. Entropy is a measure of disorder or information, and an important concept for physics. For instance, the second law of thermodynamics implies that the entropy of a system will be maxizimed over time [3]. But entropy is not just a concept in physics, it is also a key concept in information theory.

Entropy is a measure of the information density in information theory, and this we can use to our advantage. Because we can also manually approximate the information density on our own, by compressing our data and calculating the compression density. By a theory in information theory (Shannon's source coding theorem [4]) we know that this is related to the entropy (at least in the limit of large numbers). This ratio is called the Computational Information Density or CID [5], and we will further explain it and the concept of entropy in chapter 2.

To start, let us test the correspondence between CID and entropy on a simple econophysics model first. One such model is the Yakovenko model [2], in which every agent starts with the same amount of money, and at every timestep, money gets randomly distributed from one random agent to another. This may seem like an oversimplification of reality, but the equilibrium of this model is still somewhat representative of real-life wealth distribution. We will discuss this model in chapter 3, where we use it to look at the CID-entropy correspondence in simulations of this model.

A toy model is nice, but we would like to see how this correspondence holds in real economical situations. However, most traditional wealth distributions are not that available in high detail (for obvious privacy reasons), but there are now digital currencies, or Bitcoin in particular. The decentralized nature of Bitcoin allows transaction history to be public [6, 7], and using this history the CID and the entropy can be calculated at arbitrary times. A further explanation of this calculation and the results of the calculation can be found in chapter 4.

With these parameters, there is the question whether and how the CID and entropy can be predictive for the price of Bitcoin. There are some statistical methods required to find such correlations using this dataset, and those are explained in chapter 5, together with the results of these statistical methods.

Then, we conclude in chapter 6 by summarizing our results, and try to explain why the correspondence we find is not as clear as it might be, and discuss what conditions seem necessary for which uses of the CID-entropy correspondence. We finish by discussing further possible research into the subjects covered in this thesis.

Chapter 2

Computational Information Density and Shannon entropy

As stated in the introduction, we wish to consider some information-theory concepts for the Bitcoin blockchain, but first we need to introduce them. In this chapter we will introduce entropy and Computational Information Density, the most important concepts for this thesis.

2.1 An introduction to entropy

First, let us quickly introduce entropy. Entropy is a measure of information. For those with a physics background, you may recall that it is maximized in thermodynamic equilibrium. In physics, it is defined as:

$$S = -k_B \sum_i p_i \log(p_i),$$

where the summation is done over the possible microstates, and p_i is the probability of microstate i (and k_B is the Boltzmann constant). This reduces to the historical formula $S = k_B \log(\Omega)$ in the case that all $p_i = \frac{1}{\Omega}$.

For this thesis we will consider the concept of entropy in information theory instead, the Shannon entropy (or the information entropy), which is a closely related concept. The Shannon entropy is calculated of a source, that is a random variable encoded by strings. If we have some random variable X taking values in some set I, then the entropy of this random variable is:

$$S(X) = -\sum_{i \in I} p(X = i) \log(p(X = i)).$$

This is very close to the physical concept of entropy, only the Boltzmann constant is gone (and the logarithm can be of a different base depending on context). This is more useful for information-theoretical reasons, and as we are not considering physics, the Boltzmann constant will not be useful. For the rest of the thesis, this will be the entropy referred to by the word entropy (if you do not like that, assume we are working in natural units).

A particular use of entropy in information theory is Shannon's Source coding theorem, which roughly states the following [4, 8]:

Theorem (Shannon's source coding theorem (informally))**.** *N independent identically distributed variables each with entropy S can be compressed into $\frac{S}{\log(2)} \cdot N$ bits with negligible risk of information loss as $N \to \infty$, conversely if they are compressed into fewer then $\frac{S}{\log(2)} \cdot N$ bits it is almost certain information will be lost.*

In other words, if you have many copies of a random variable X, the best possible lossless (with arbitrarily high probability as N increases) compression for the data results in $\frac{S(X)}{\log(2)} \cdot N$ bits[1]. This theorem connects the concept of entropy to the concept of compression, which is crucial to the next concept, the CID.

[1] The theorem does not say that this compression rate is possible, but any compression worse then this rate is

2.2 An introduction to CID

Now, the entropy is useful in this thesis in an information-theoretical way, through Shannon's source coding theorem, which relates it to compression. This theorem suggests another way of calculating the entropy: by using traditional compression algorithms and calculating the rate of compression, the Computable Information Density (CID). As a formula:

$$\mathrm{CID} = \frac{\text{number of bytes compressed}}{\text{number of bytes uncompressed}}.$$

This means we would be compressing the raw data that comes from a model to approximate the entropy. So, if the model were to produce a datafile of 4 GB, and compressing it would turn it into a datafile of 1 GB, the CID would be $\frac{1}{4}$. This can be especially convenient, because if it is necessary for the entropy to make sense to consider many different simulations (to extract a probability distribution), we only need one result to define the CID.

There have been a few studies in physics about the CID-entropy correspondence already (for instance [5]), but for this thesis, we will study the correspondence in econophysics.

Chapter 3

The Yakovenko Model

This chapter discusses the Yakovenko model [2], a simple agent-based econophysics model, to use it to test the CID-entropy correspondence. We give a description of the model and find the equilibrium in two different ways, and compare those exact equilbria to a simulation of the model. Consequently we use this simulation to study the CID-entropy correspondence.

3.1 The Yakovenko Model explained

Let us begin by introducing the Yakovenko model. It is a very simple model that models a society in which money is randomly exhanged. This model actually predicts something close to the real-world distribution of income (for the poorest 95%) and can even be solved analytically, so it is a good toy model for us. In the next few sections, we will introduce this model and solve it.

The Yakovenko model consists of a system which has N agents, each with their own money; let us say agent i has money $(m_i)_t$ at time-step t. To start, every agent begins with the same amount of money $m_0 = (m_i)_0$, and then the following steps are taken (for each timestep t):

- Two agents i_t, j_t are selected at random, with $i_t \neq j_t$
- An amount of money $\Delta_t \in [0, \Delta_{max}]$ is chosen uniformly at random
- If agent i_t has at least Δ_t money ($(m_{i_t})_t \geq \Delta_t$), then agent i_t gives Δ_t money to agent j_t;

We can write this mathematically as:

$$\begin{aligned}
(m_{i_t})_{t+1} &= (m_{i_t})_t - \Delta_t : & (m_{i_t})_t &\geq \Delta_t \\
(m_{j_t})_{t+1} &= (m_{j_t})_t + \Delta_t : & (m_{i_t})_t &\geq \Delta_t \\
(m_k)_{t+1} &= (m_k)_t : & k \neq i_t, j_t \text{ or } (m_{i_t})_t &< \Delta_t
\end{aligned}$$

The model may seem very simple, yet it does produce some nice results.

3.2 Using the Fokker-Planck equation to calculate the equilibrium

We can find an equilibrium for the model by looking at the Fokker-Planck equation, which we can derive in the following way: If we consider some infinitesimal change of the probability distribution $dP(m)$, probability flow towards this is caused by Δ money flowing from $m + \Delta$ to $m' - \Delta$ summed over all m' (to get m, m'). As a formula, this term is:

$$\int_0^\infty dm' \int_{-\Delta_{\max}}^{\Delta_{\max}} d\Delta P(m+\Delta)P(m'-\Delta)R(m+\Delta, m'-\Delta, \Delta)dt,$$

where $R(q,q',\Delta)$ is the flow rate of Δ money from q to q'. Money flow away (from m,m') is caused by the opposite effect, yielding a term:

$$-\int_0^\infty dm' \int_{-\Delta_{\max}}^{\Delta_{\max}} d\Delta P(m)P(m')R(m,m',-\Delta)dt.$$

Now, this is a detailed balance: $R(m+\Delta,m'-\Delta,\Delta) = R(m,m',-\Delta)$, as $m+\Delta > \Delta$ precisely when $m > 0$ and $m'-\Delta > 0$ precisely when $m' > \Delta$, and because rate does not depend on Δ and m,m' otherwise. In other words, these rates are equal because the transactions are possible if and only if the reverse transactions are possible, and all strictly positive rates are equal in this model[1]. So our Fokker-Planck equation looks like:

$$\frac{dP(m)}{dt} = \int_0^\infty dm' \int_{-\Delta_{\max}}^{\Delta_{\max}} d\Delta R(m,m',-\Delta)(P(m+\Delta)P(m'-\Delta) - P(m)P(m')).$$

To calculate an equilibrium, we require that the first term in the integral cancels to the second term in the integral, to set the derivative to 0. This gives rise to equations:

$$P(m+\Delta)P(m'-\Delta) = P(m)P(m'),$$

for all $m,m' > 0$ and for all Δ, implying that $P(m) = Ae^{Bm}$ (this we can derive from $P(m+\Delta)/P(m)$ being independent of m for all Δ). Now, we also have the constraint that the distribution has to be normalized, so:

$$1 = \int_0^\infty dmP(m) = \left[\frac{A}{B}e^{Bm}\right]_0^\infty.$$

So $B < 0$ and $\frac{A}{B} = -1$. The other constraint we have is that the average money is constant (as the total amount of money is), so:

$$m_0 = \langle m_i \rangle = \int_0^\infty dmP(m)m = -\int_0^\infty dmBe^{Bm}m.$$

We can calculate this integral through partial integration:

$$-\int_0^\infty dm\frac{1}{B}e^{Bm}m = -\left[Be^{Bm}m\right]_0^\infty + \int_0^\infty e^{Bm}dm = -\frac{1}{B},$$

giving $P(m) = \frac{1}{m_0}e^{-\frac{m}{m_0}}$, the Boltzmann distribution.

3.3 Using entropy maximalization to calculate the equilibrium

We can also find an equilbrium for the model by maximizing the entropy. This will also result in an equilibrium, as in thermodynamic equilibrium entropy is maximized. Just maximizing the entropy with a functional derivative is not sufficient, however, as there are constraints on our system, so we will need to use Lagrange multipliers.

The first restriction comes from the assumption that the distribution is normalized: $\int_0^\infty dmP(m) = 1$. We also presume the total amount of money is conserved, which translates in our integral notation that the average is conserved (as the total amount of money is not as well defined with a smooth distribution P). This results in the restriction $\int_0^\infty dmP(m)m = m_0$. Applying the method of Langrange multipliers gives the following equation:

$$\frac{\delta}{\delta P(m)}\left(-\int_0^\infty dm'P(m')\log(P(m')) + \lambda\left[\int_0^\infty dm'P(m') - 1\right] + \mu\left[\int_0^\infty dm'P(m')m' - m_0\right]\right) = 0.$$

Solving this gives $-\frac{P(m)}{P(m)} - \log(P(m)) + \lambda + \mu m = 0$, so $P(m) = e^{-1+\lambda+\mu m}$. As this is the same as $P(m) = Ae^{Bm}$ like the previous section, filling in the solutions to the boundary conditions found allows us to derive that $P(m) = \frac{1}{m_0}e^{-\frac{m}{m_0}}$, which is again the Boltzmann distribution. [2]

[1] Only that they are equal for the same Δ is required for this argument to work.

[2] Note that this derivation did not use any properties of the way the amount of money to be exchanged is chosen (except that it is the same for every agent), so this derivation shows that different similar models can have the Boltzmann distribution as an equilibrium.

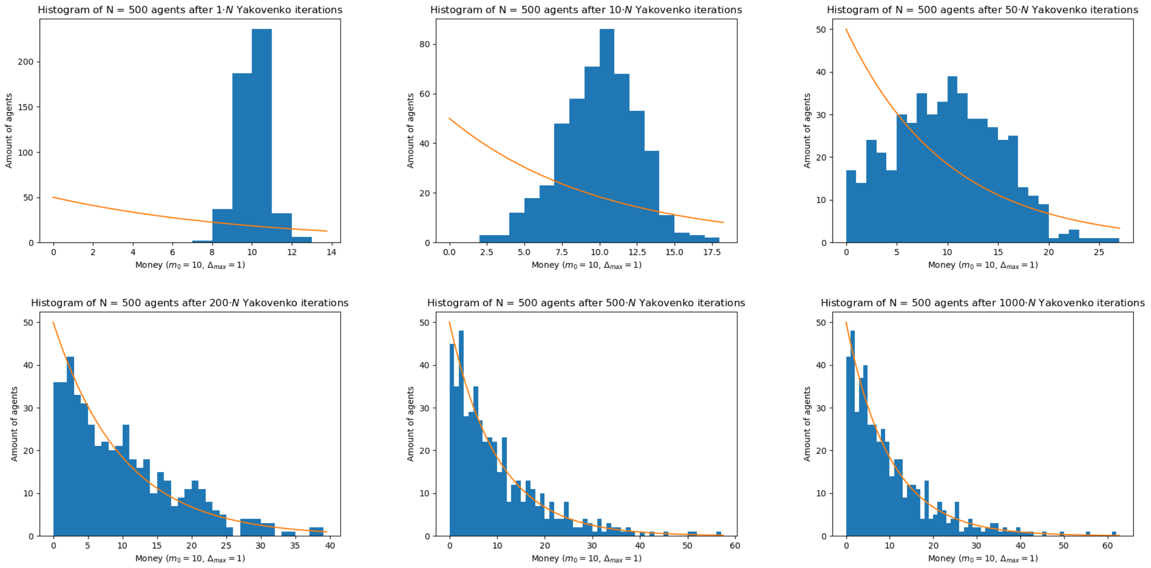


Figure 3.1: The distribution of money for the Yakovenko model ($m_0 = 10$, $\Delta_{\max} = 1$) after N, $10 \cdot N$, $50 \cdot N$, $200 \cdot N$, $500 \cdot N$ and $1000 \cdot N$ iterations (the y-axis displays the number of agents with that amount of money). We can see the Boltzmann distribution, the equilibrium we found in the previous section, as a comparison, and we can see that the Yakovenko model does converge to it.

3.4 CID and entropy for the Yakovenko model

So now that we have a simple model to use, let us analyze the CID as a proxy for the entropy by calculating it for the Yakovenko model and comparing it to the entropy. A numerical simulation of this model was run using Python (my code can be found at [9]), the results of which we can see in figure 3.1. We see in this figure the money distribution of the simulation after different numbers of iterations. As the number of iterations increases, we can see how the distribution converges to the equilibrium we calculated in sections 3.2 and 3.3, the Boltzmann distribution. This suggests that the model works as expected, so let us try to analyse the CID and entropy here.

To analyze the effectivity of the CID as a primer for the entropy for this model, multiple compression libraries were used to calculate the CID (zlib [10], gzip [10], lz4 [11] and lzma [12]). The CID was calculated by compressing a snapshot of the simulation. This data is created by taking the money of each agent, rounding it to integers, and then taking the binary string of the numbers with the money of each agent. For example if there were three agents, with the first having 100 money, the second having 011, and the third having 101 money, the bitstring to be compressed would be 100011101[3].

Note that binning the data after the simulation of the model is required for a good CID calculation, as the Yakovenko model is continuous and we do not have infinite datapoints. So the physical information is in the whole number, but floating point numbers store a constant amount of bits and the order size. This means any direct compression of floats would be unphysical[4], and therefore the binning is necessary[5].

We can see the CID approximately converges to a constant value in figure 3.2, where we can see the CID over many iterations in the system. Note this convergence to a small distribution around a mean, which is not that strange, as small variations will still happen around the equilibrium. As the CID only approximately converges to a mean, we take the average of multiple samples of the CID after a certain amount of iterations[6] to better approximate the actual entropy.

[3] Of course, 3-bit numbers weren't used, this is a simplification.

[4] Unless all the floating point numbers are between 2^k and 2^{k+1} for some whole number k, which isn't the case for the Yakovenko model.

[5] An alternative to this would be to consider a discrete distribution instead of a continuous distribution (which results in the same equilibrium, and is a very similar model).

[6] N iterations, to be precise.

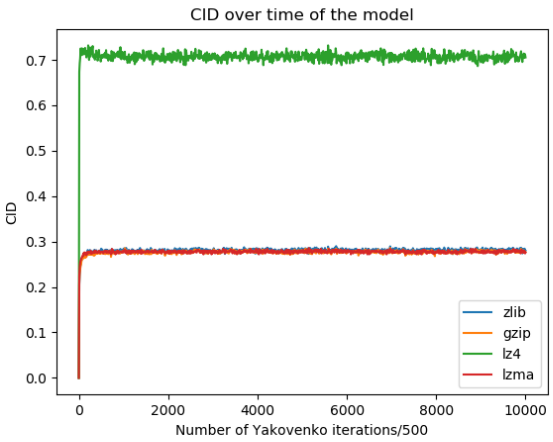


Figure 3.2: The CID of the Yakovenko model as a function of the number of iterations, calculated as described in section 3.4. We see a convergence to something that seems to be a random variable around a mean.

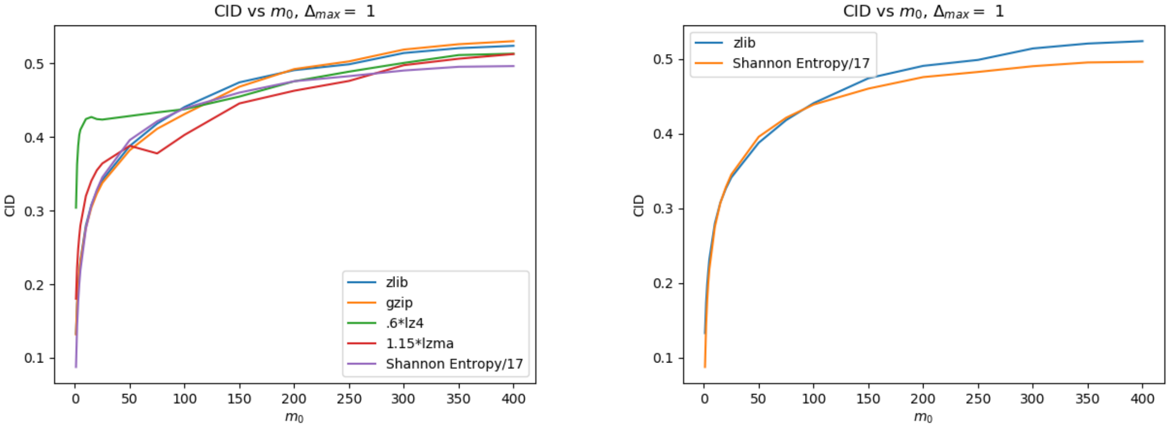


Figure 3.3: The CID plotted versus the initial money, with as a comparison the scaled Shannon entropy. The CID is calculated after 100 000 times $N = 500$ iterations, after which the values of the CID have converged to the values at the equilibrium. Note that the CID found by using the zlib algorithm seems to be a very good measure of the entropy.

We tried to calculate the CID as a function of m_0 (which is the temperature of the Boltzmann distribution in the equilibrium, as derived in sections 3.2 and 3.3) after $100000 \cdot N$ iterations, to get close to thermodynamic equilbrium. As a comparison the explicit entropy of the distribution as a function of m_0 is calculated as well (for the entropy, the binning is also done by rounding to integers to extract a distribution). The result of this is shown in figure 3.3, where we can see these functions, rescaled to have similar shapes.

We can see in figure 3.3 that whilst there are multiple algorithms that potentially can give a CID that is a good approximation of the entropy, zlib does seem to be one of the best [7]. So for Bitcoin, we decided to just stick to zlib, as these calculations take quite some time both to implement and to run.

[7] The second best would seem be gzip, which is effectively zlib but with a different header.

Chapter 4

Entropy and CID for Bitcoin

Now, it is time to apply the techniques we have learned by looking at the simple Yakovenko model to a real-world scenario. However, looking at real money distributions is hard, as these are not freely accessible. Bitcoin on the other hand has by nature a public record, because it uses a blockchain [6]. That means roughly that the transactions are stored in blocks, and that these blocks refer to each other in order to guarantee that the transactions are correct. It is a complex process, but these transactions are available.

This means that by looking at the blockchain and following the transactions we can analyse the distribution of Bitcoins exactly, and that way extract the distribution. In this thesis we will use the dataset from [13] (and [7]), which consists of all Bitcoin transactions up to 2018[1] (which will allow us to infer the distribution at any time up until then).

4.1 Bitcoin dataset

Now, how do we calculate the distribution at arbitrary times from this dataset? Let us inspect the dataset from [13].[2]

First there is the file ***bh.dat***, which has a line per block. This file has 4 columns: ***blockID, hash, block_timestamp, n_txs***. These columns are the blockID used in this data format, the blockchain hash, the timestamp, and the number of transactions of that block. We use this file to assign to every block a timestamp, so we only look at the first and third column.

The second file we use ***tx.dat***, which has a line per transaction. This file has 4 columns: ***txID, blockID, n_inputs, n_outputs***. These columns are the transaction ID(txID), the blockID, the numper of inputs and the number of outputs of the transaction. We use this file to assign to every transaction a block (so through ***bh.dat*** a timestamp), so we only look at the first and second column.

Now we get to the important files, first ***txout.dat***, which has a line for every output(credit) of a transaction. This file has 4 columns: ***txID, output_seq, addrID, sum***. These columns are the transaction ID, the output sequence (the position in the number of outputs of this transaction), the adress ID, and the sum of Satoshis (10^{-8} Bitcoin) being credited for each output. We use this file to apply the credit transactions, so we only look at the first, third and fourth column.

And last ***txin.dat***, which has a line for every input (debit) of a transaction. This file has 6 columns: ***txID, input_seq, prev_txID, prev_output_seq, addrID, sum***. These colums are the transaction ID, the input sequence (position in the number of inputs of this transaction), the transaction ID of the previous transaction this address got credited, the output sequence of the previous time this adress got credited, and the sum in Satoshis being debited for each input. We use this file to apply the debit transactions, so we only look at the first, fifth and sixth column. Note that not every transaction has an input, because Bitcoins can be generated by mining.

Now, knowing this, we can evaluate the transactions at the blockchain to get the distribution at arbitrary times by looping over blocks. For each block, we loop over the transactions associated with the block. For each transaction, we loop over all inputs and credit the associated address ID, and we loop over all outputs and debit

[1] As the CID calculation slows down significantly as the system size increases, this is more data then my pc can calculate the CID of in reasonable time.

[2] The data in the blockchain is very similar but slightly differently formatted.

the associated address ID. This way we can produce the distribution at any time, and using this we can calculate the CID and entropy at any time. This explanation should make it simple to reproduce or read the program we made to calculate the CID and entropy (for my code, see [9]).

4.2 Differential Entropy calculation

One of the reasons to consider CID is because it is a lot quicker at giving a result for the entropy then the explicit entropy calculation itself. However, this convenience complicates this thesis as well, precisely because we wish to compare the CID to the entropy. For the Yakovenko model, it was still feasible to extract the entropy at every time from the distribution at that time (i.e. calculate $-\sum_i p_i \log(p_i)$ from the distribution of every timestamp), as that model does not involve too large of a dataset. However, the Bitcoin blockchain is so much larger that this method is rather slow. Instead, as we have access to the differences, calculating the entropy difference compared to the previous timestamp is actually a lot easier, if we consider it in the following way: (where $p_i = k_i/n$ is the probability of having i Bitcoins):

$$\begin{aligned} S(t_{j+1}) - S(t_j) &= -\sum_i (p_i(t_{j+1}) \log(p_i(t_{j+1})) - p_i(t_j) \log(p_i(t_j))) \\ &= \log(n(t_{j+1})) - \log(n(t_j)) - \sum_i (\frac{k_i(t_{j+1})}{n(t_{j+1})} \log(k_i(t_{j+1})) - \frac{k_i(t_j)}{n(t_j)} \log(k_i(t_j))). \end{aligned}$$

So, with only the change in the total amount of agents n and the change of the individual bins k_i (which is exactly what the dataset is like as seen in the previous section) we can easily calculate the change in entropy, this way giving us a convenient way to calculate the entropy quickly.

4.3 Results

Now that we know how to get the data and calculate the CID and entropy, we can see how the CID[3] (for the code see [9]) and entropy develop over time, this is plotted in figure 4.1. We see, as we expected, that the CID is larger then the entropy[4]. We see a clear correspondence between both measures of information at the larger scale, let us inspect some features to see if this correspondence still holds at smaller scales.

We will start by looking at the valley in December 2009 and January 2010, when Bitcoin was still young, in figure 4.2a. We see large rough spikes in the entropy and corresponding spikes in the CID. There are approximately straight lines between those, which are a lot more noisy for the CID but still similar. This kind of noise we can expect based on the noise in the CID we found for the Yakovenko model (see section 3.4). Note that whilst the functions are similar, they are not exactly a function of eachother (although close to one).

Let us study some more of these features of these graphs, to see how similar they are in detail and to see if we can find real-world explanations for the effects. Consider the significant increase of both CID and entropy in mid 2010. A detailed plot has been made in figure 4.2b of this sharp increase in further detail. We can see for both the entropy and the CID that there is a strong increase in the early hours of July 12th, at almost the same time. Looking into the history of Bitcoin, this could be explained by a popularity and price spike coming from a popular article on Bitcoin of the previous day [14].

Once again this is not the only feature which is similar, the smaller features are still visible and very similar for both the CID and the entropy, even when looking closer at the graph. We no longer see the smaller changes in CID as we saw at the end of 2009 and with the Yakovenko Model, as every feature of the CID seems to correspond to another feature of the entropy; this is probably caused by the increase in agents (Bitcoin becoming more popular) causing random changes of the CID to be relatively smaller.

Now, let us inspect another feature, the peak of this increase at mid 2011, plotted in figure 4.2c. This peak corresponds to a top of the first Bitcoin price bubble at 8 July 2011 [5]. We again see very similar features,

[3] In this section, we are actually looking at the CID $\cdot \log(2^{64})$, as $\frac{S}{\log(2^64)}$ is the maximum compression ratio if the amount of bitcoin for different agents is identical, as we can derive from Shannon's source coding theorem, see chapter 2.

[4] We could theoretically see a small amount of time where the CID is lower then the entropy, which is something you may expect to be forbidden under the Shannon source coding theorem (see chapter 2). However, the Shannon source coding theorem only works for a random and independent source, and this is inferring the probability distribution from the outcome distribution instead, which means that there might be intercorrelations.

[5] This bubble was popped by a huge security breach in one of the bigger exchanges at the time.

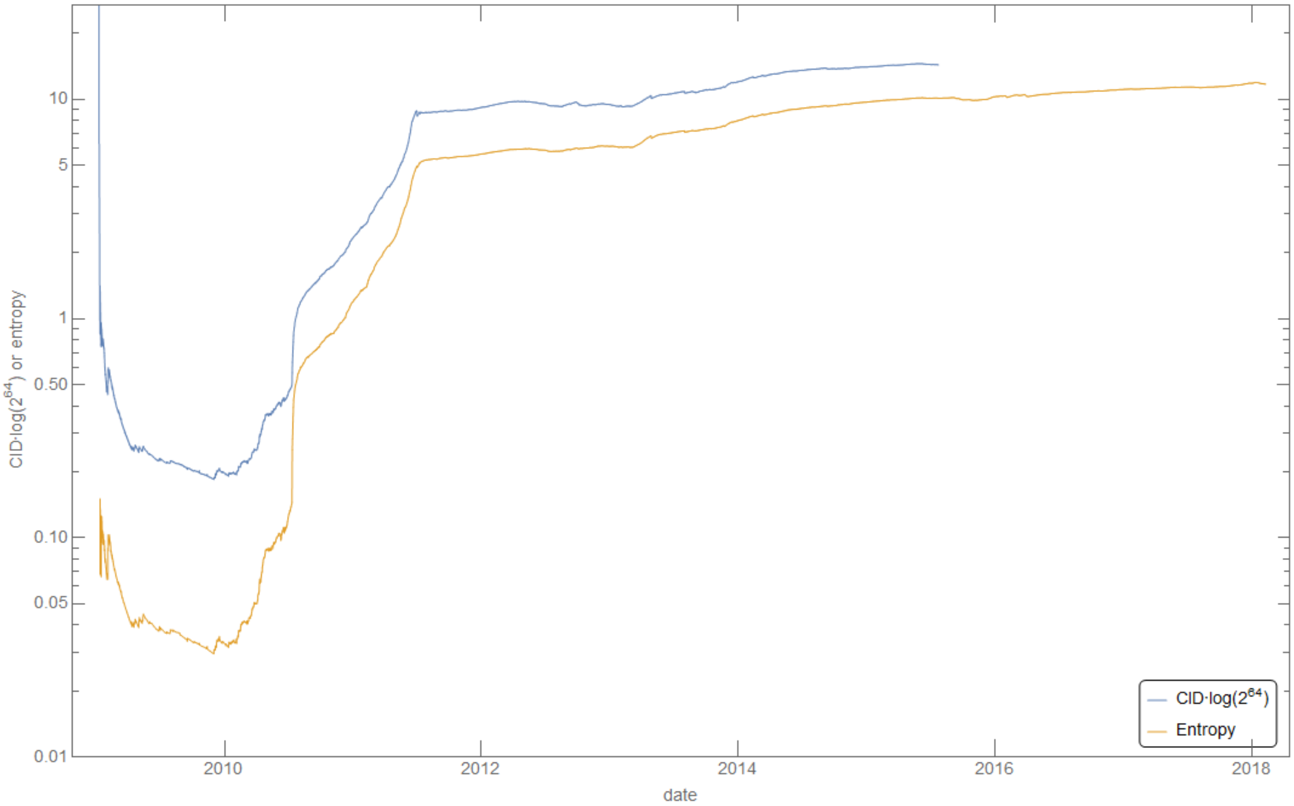
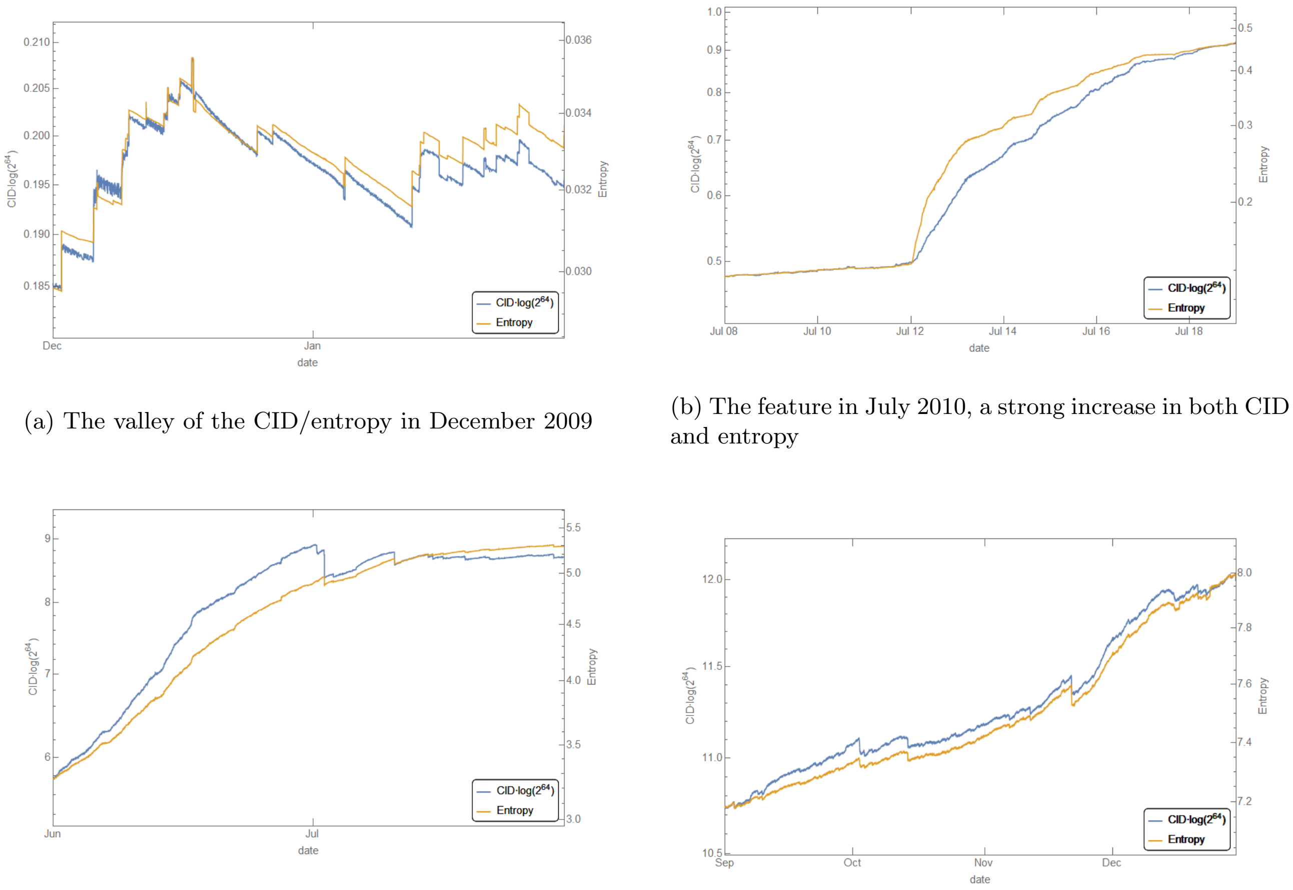


Figure 4.1: The CID and entropy of the Bitcoin blockchain as a function of the date. We can see very similar features and they have a very similar shape.

although some of the changes in the CID are significantly larger compared to the changes of the entropy. The day-by-day features are again very similar, as we have come to expect, but there are again changes which are not exactly the same.

Finally, let us look at a later part of the timeline, the last 4 months of 2013. In figure 4.1 it may seem like there are little obvious features, but that is mainly because the entropy starts to stabilize. If we look at figure 4.2d we can see that this is not the case, that the two graphs still have clear features, which are very similar. As Bitcoin was growing more popular, quite a bit happened in these months: in November the price increased strongly, from around \$200 to \$1242 at the end of November, after which there was another crash and then stabilization.



(a) The valley of the CID/entropy in December 2009

(b) The feature in July 2010, a strong increase in both CID and entropy

(c) The feature in mid 2011, a strong peak of entropy and a weak peak of CID, after which the CID overtakes the entropy again

(d) The CID and entropy at the end of 2013

Figure 4.2: Four graphs, each representing a part of the timeline, showing the development of entropy and CID of the blockchain, as discussed in section 4.3. The figures shown are rescaled to make the entropy and the CID have the same scale, so the y-axis is different for the CID as compared to the entropy (the CID axis is on the left, the entropy on the right). Take note of the small scale features of both measures of information, which are very similar when compared to eachother, but not identical.

Chapter 5

Bitcoin price correlations

With these measures, which seemingly somewhat correlate to real-life events, a question that comes to mind is: “Can we predict the Bitcoin price based on either the entropy or the CID?” If we could, then investing would allow us to gain a lot of money, and it would also further validate the use of econophysical techniques (of course, the second one is more important).

5.1 How to calculate the price

Now, this data is can be useful outside of theoretical context because we have something to use it for: to consider the Bitcoin prices. However, there is no unique Bitcoin price, as there are many exchanges with different prices [15], so there is no obvious measure to calculate the price. While we can download the raw data (see [16]) of different exchanges with different prices and different volumes, we still need to find a way to turn this data in a price graph over time. A naive way to do this would be to consider just the last price being used, but this has a lot of flaws. For instance, if there is a small amount of Bitcoins sold for a significantly different price in the last transaction, then this price does not need to be representative of the current valuation of Bitcoin. So we would like our price calculation to consider some of the prices of past transactions as well.

So instead we consider the following: Let p_i be the price of the i-th transaction at time t_i with volume v_i. Then we set the price at time t to be equal to $P(t)$:

$$\begin{aligned} pv(t) &= \sum_{i:t_i \le t} p_i v_i e^{-\frac{t-t_i}{\tau}} \\ v(t) &= \sum_{i:t_i \le t} v_i e^{-\frac{t-t_i}{\tau}} \\ P(t) &= \frac{pv(t)}{v(t)}, \end{aligned}$$

with τ being some typical time. Note that we defined this by having a time-corrected (Bitcoin) volume $v(t)$ and a time-corrected price×volume (or dollar volume) $pv(t)$. In figure 5.1 this is explained in a visual way. In this figure we see when Bitcoins are purchased at a particular price by a circle, with the surface area of the circle representing the amount of Bitcoin bought. We can see that small Bitcoin purchases only affect the price when there have been little other purchases around this time.

Note that as $\tau \to 0$ we return to the case in which we just consider the last price being used, as $pv(t)e^{\frac{t-t_i}{\tau}} \to p_l v_l e^{-\frac{t-t_i}{\tau}}$ and $v(t)e^{\frac{t-t_i}{\tau}} \to v_l$ where $t_l \le t < t_{l+1}$ (and as $\tau \to \infty$ we get the average price up until then). In the case $\tau - > \infty$ we get to the average of all previous prices instead. Also note that this is still not smooth; a way to make this continuous would be to consider the future prices as well in some particular way[1], but this seems nonphysical.

In this paper, we used this method to calculate the price at particular times. Ref. [16] provides us with a list of different exchanges, but those are also in different valuta. So we used [17] to convert all prices to USD. The

[1] An example of C^∞ price calculation would have as weight $e^{-\frac{(t-t_i)^2}{\tau^2}}$, for all i.

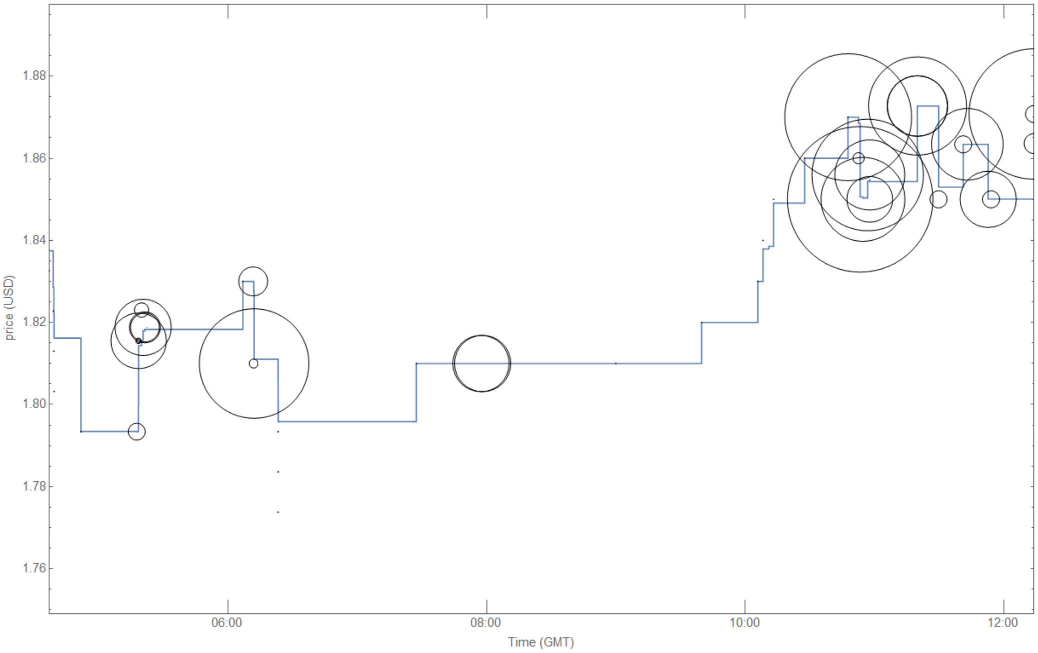


Figure 5.1: A visual representation of the price calculation, on a sample dataset. In this figure circles indicate transactions, with the surface area of the circle scaling with the price and the center of it the price/time. As argued, locally small transactions seem to make little difference.

data in [16] is unreliable, however, because there is data in there with negative volume and/or negative price, or with incredible spikes in price (for instance, some data indicates a price of 0). So we check that the price doesn't change by a factor of 100 and that the volume is positive in order to remove some of the junk data.

5.2 Correlation time

So with these measures, we might wonder about any predictive power of the CID or the entropy for the price. In order to test whether there is any predictive power, we might at first consider the following coefficient (for the entropy here, but similarly for the CID[2]):

$$\begin{aligned} r_{S(t),\log(\$(t+\Delta))} &= \left\langle \frac{S(t)-\langle S(t)\rangle}{\sigma_S} \cdot \frac{\log(\$(t+\Delta)) - \langle \log(\$(t+\Delta))\rangle}{\sigma_{\log(\$)}} \right\rangle \\ &= \frac{\langle S(t)\log(\$(t+\Delta))\rangle - \langle S(t)\rangle\langle \log(\$(t+\Delta))\rangle}{\sigma_S\sigma_{\log(\$)}}, \end{aligned}$$

where in our case the averages are taken over the time t (because we have nothing else to average over, unfortunately). We are looking at the logarithm of the price because whilst the entropy tends to stay in somewhat same order sizes, the price does not, so any correlation considering just the price would be highly dominated by the last part of the Bitcoin blockchain.

This measure is called the Pearson correlation coefficient or the Pearson r (sometimes called the PCC). It is a value between 1 and -1, where 1 is a linear relation with positive coefficient between the entropy and the price (i.e. $S(t) = b\log(\$(t+\Delta))$ for some $b > 0$), and -1 a linear relation with negative coefficient (i.e. $S(t) = b\log(\$(t+\Delta))$ for some $b < 0$).

However, there is an issue here; both the entropy and price increase over time. And as our average is not over a probablistic distribution, but a distribution over time, we expect our correlation $r_{S(t),\log(\$(t+\Delta))}$ to not decrease over time. After all, as the price is higher in the future in general, we expect the price to be higher for

[2] The correlation coefficients in this section are the same for CID as for CID $\cdot \log(2^{64})$, so we measure the predictive power of both.

higher Δ, so the Pearson r is higher for higher Δ. So no new information would come from evaluating this, we would mainly see that the price of Bitcoin increases and so do these measures of information.

So instead, we are considering:

$$r_{\frac{dS}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)} = \left\langle \frac{\frac{dS}{dt}(t) - \langle\frac{dS}{dt}(t)\rangle}{\sigma_{\frac{dS}{dt}}} \cdot \frac{\frac{d\log(\$)}{dt}(t+\Delta) - \langle\frac{d\log(\$)}{dt}(t+\Delta)\rangle}{\sigma_{\log(\$)}} \right\rangle$$

$$= \frac{\langle\frac{dS}{dt}(t)\frac{d\log(\$)}{dt}(t+\Delta)\rangle - \langle\frac{dS}{dt}(t)\rangle\langle\frac{d\log(\$)}{dt}(t+\Delta)\rangle}{\sigma_{\frac{dS}{dt}}\sigma_{\frac{d\log(\$)}{dt}}}.$$

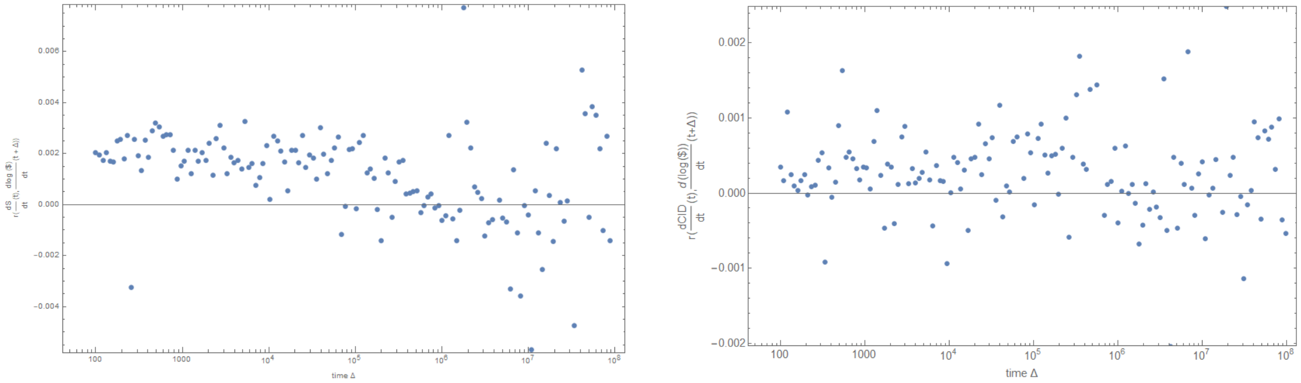
This is the Pearson r of the derivatives[3]. Considering the derivatives could say how much a change in entropy is predective for a change in price.

5.3 Results

We calculated the CID and entropy difference correlations with the future price difference, to see if either has any simple predictive property for future price changes. In figure 5.2a we can see this function for the entropy, where we do see some correlation. This correlation seems to be predictive for a bit but fall off over time. This decrease over time is good, because that means we have found a measure that is not just predictive because both consistently increase over time. However, for the CID we see no such effect in figure 5.2b.

To inspect the correlations found for the entropy, we need something to compare it to. Let us look at the auto-correlation function of the price difference, to see how predictive the price is for the future price. We can see this function in figure 5.3. We see a much stronger correlation for all time differences, so unfortunately the entropy is not more predictive of the future price then the price itself, although that is not that surprising. However, we don't know yet if the correlations of the entropy can be considered significant, because whilst the price may be more important for the future price, the entropy could add a bit of information.

So in order to inspect the correlations with the future price, let us consider the Pearson r of the entropy difference with the current price difference. This term indicates how much the current entropy difference is predictive for the current price difference, if we combine this with the Pearson r of the current price difference with the future price difference we can see if the entropy does anything extra. We can see this comparison in figure 5.4. This comparison suggests there is some predictive power in the entropy change for the future price change, as these shapes are dissimilar. Keep in mind though, that this linear effect is only of the size $r = 0.002$, as compared to the predictive power of the price difference of close to $r = 1$.



(a) $r_{\frac{dS}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)}$ as a function of Δ

(b) $r_{\frac{d\text{CID}}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)}$ as a function of Δ

Figure 5.2: Pearson r of the derivative of the entropy/CID and the derivative of future price, as a function of the time difference. Note that we only see an effect for the entropy, not for the CID.

[3] With derivatives we here mean the approximations $\frac{dS}{dt}(t_i) = \frac{S_i - S_{i-1}}{t_i - t_{i-1}}$ and $\frac{d\log(\$)}{dt}(t_i + \Delta) = \frac{\log(\$(t_i+\Delta)) - \log(\$(t_{i-1}+\Delta))}{t_i - t_{i-1}}$ where t_i is the time of the ith block and S_i the entropy of it.

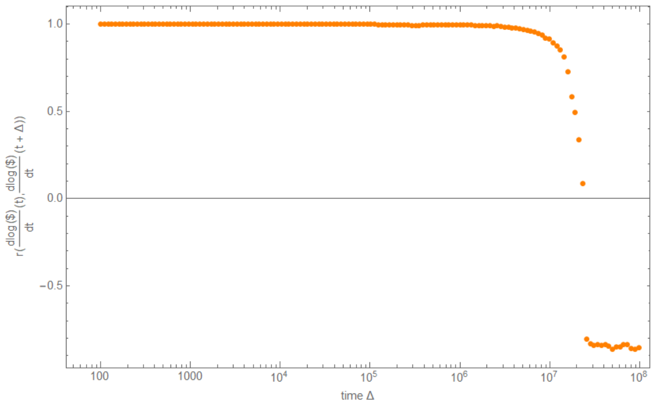


Figure 5.3: A plot of the Pearson r of the derivative of the price with the derivative of the future price($r_{\frac{d\log(\$)}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)}$) as a function of delta. Note how strong the effect is.

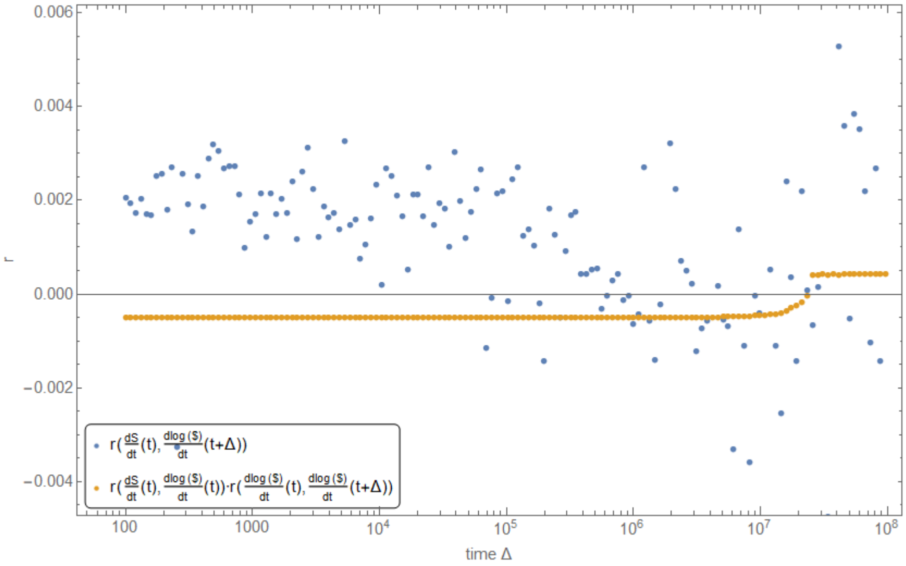


Figure 5.4: $r_{\frac{dS}{dt}(t),\frac{d\log(\$)}{dt}(t)} \cdot r_{\frac{d\log(\$)}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)}$ (orange) compared to $r_{\frac{dS}{dt}(t),\frac{d\log(\$)}{dt}(t+\Delta)}$ (blue) as a function of Δ. This comparison suggests that the entropy difference has some extra predictive power for the future price beyond the predictive power for the current price, given the dissimilarity of the graphs.

Chapter 6

Conclusion

6.1 Summary and conclusion

In this thesis we discussed the concepts of entropy and CID. We illustrated these concepts by calculating them for a simple econophysics model, the Yakovenko model. We first introduced and found the equilibrium of this simple model, and then performed numberical simulations which converged to this equilibrium. We used these simulations to calculate the entropy and CID, and found the expected correspondence for some of the chosen algorithms. In this part we learned an important lesson: the data type of the data is important, as floating point numbers are stored in a way that causes nonphysical binning of data.

One of the algorithms was picked and used to make this comparison for the Bitcoin blockchain, where we still found a good resemblance. We calculated the entropy in a differential way to more quickly calculate it, to make the comparison with CID feasible. We can again identify similar features between the CID and entropy, but they do vary in relative size. We identified some of the features and tried to connect them to real-life events.

Finally, we tried to use this data to see if the entropy or CID had some predictive power. We did this by calculating linear correlation coefficent (the Pearson r) of the derivative of the entropy with the derivative of the price at a future time (the derivative was necessary to avoid the conclusion that the future price is higher for higher entropy, which we already know as both have increased over time). We found a small effect, but not nearly as large as the predictive power of the price derivative for the future price derivative.

6.2 Discussion and outlook

This thesis illustrated why Bitcoin may be used as an interesting model for econophysics: the microscopic data is freely available. It is not often that we get access to precise data at such a scale. It can be used as a nice toy model for econophysics very conveniently. Bitcoin can be considered interesting because it goes through a lot of system size scale changes, and we saw this in the noise in the CID not being visible when the system grew large enough. This caused the CID to be a strong measure for the entropy, with nearly all features being similar.

The CID-entropy correspondence was strong, but not one-to-one. This could be caused by patterns in data, for instance that the amount of bitcoin of two agents next two each other is correlated. As the compression algorithms looks for patterns in the entire dataset, these correlations could change the compression tactics, causing the CID to be lower. But for the entropy, no order is considered, only a distribution, so these effects are not visible. A way to negate this could be to first put the data in random order, but this would cause the CID to vary a lot. To decrease this variance, an average over multiple random orders could be taken (but this might not be faster then the explicit entropy calculation).

It might be interesting to consider this difference between the CID and the entropy in further detail. An exact analysis of the CID-entropy correspondence could be done, where the compression algorithms are no longer treated as black boxes, but as known algorithms. Another way to to consider this difference is as a measure of larger features of the system. For instance, the analysis which was done for the CID and the entropy as predictive for the price could be done for some measure of this difference as predictive for the price, maybe this could result in a stronger correlation.

Another analysis that could be done is to connect the entropy and CID to the price in further detail as well, going for a nonlinear correlations or just looking into them into a further detail, to find a way to practically use this correlation, as our analysis only suggested a small correlation. Other analysis that could be explored is to study the features of the entropy and CID by hand and analyse how they connect to the price, as only some features have been studied in this thesis. This could also be instrumental to a further analysis of the entropy-CID connection.

To conclude, we found that the Bitcoin blockchain proves itself to be a powerful toy model for econophysics in this thesis. And that the concept of CID shows a ton of promise to provide a measure for the entropy.

Bibliography

[1] Mantegna R.N., Stanley H.E., An Introduction to Econophysics, Cambridge University Press (2000), ISBN 0 521 62008 2

[2] Dragulescu, A., Yakovenko, V., Statistical mechanics of money, Eur. Phys. J. B 17, 723729 (2000) https://doi.org/10.1007/s100510070114

[3] Feynman R., Leighton R.B., Sands M., The Feynman Lectures on Physics, Volume I, Basic Books, ISBN 978-0-465-02414-8, https://www.feynmanlectures.caltech.edu/

[4] Shannon C.E., A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, pp. 379423, 623-656, July, October, 1948

[5] Martiniani S., Chaikin P.M., Levine D., Quantifying Hidden Order out of Equilibrium, Phys. Rev. X 9, 011031, (2019) https://doi.org/10.1103/PhysRevX.9.011031

[6] Nakamoto S., Bitcoin: A Peer-to-Peer Electronic Cash System (2009), Retrieved from https://bitcoin.org/bitcoin.pdf Accessed 26 July 2020

[7] Kondor D., Psfai M., Csabai I., Vattay G., Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network, PLOS ONE 9(2): e86197. (2014), https://doi.org/10.1371/journal.pone.0086197

[8] MacKay, D.J.C., Information Theory, Inference, and Learning Algorithms, Cambridge University Press, (2003), ISBN 0-521-64298-1

[9] Nesenberend, C.T., “btcCIDEnt”, https://github.com/ctnesenberend/btcCIDEnt

[10] Gaily J., Adler M., “zlib Home Site”, Retrieved from https://www.zlib.net/ Accessed 26 July 2020

[11] “LZ4 - Extremely fast compression”, Retrieved from http://www.lz4.org/ Accessed 26 July 2020

[12] “LZMA SDK (Software Development Kit)”, Retrieved from https://www.7-zip.org/sdk.html Accessed 26 July 2020

[13] Kondor D., “Bitcoin network dataset”, Retrieved from https://senseable2015-6.mit.edu/bitcoin/ Accessed 26 July 2020

[14] Dawson, K., “Bitcoin Releases Version 0.3 - Slashdot” (11 July 2011), Retrieved from: https://slashdot.org/story/10/07/11/1747245/Bitcoin-releases-version-03 Accessed 26 July 2020

[15] Bitcoincharts, “Bitcoincharts”, Retrieved from https://bitcoincharts.com/markets/ Accessed 26 July 2020

[16] Bitcoincharts, “Index of /v1/csv/”, Retrieved from: https://api.bitcoincharts.com/v1/csv/ Accessed 26 July 2020

[17] Antweiler W., “Pacific Exchange Rate Service - Database Retrieval System”, Retrieved from: http://fx.sauder.ubc.ca/data.html Accessed 26 July 2020