Methodology behind the statistical engine of CromoLab platform. Trend analysis, portfolio construction and hedging with advanced econometrics

CromoLab

11, 14, 2017

Abstract

CromoLab creates statistical and technological infrastructure necessary for effective investing in cryptocurrencies. CromoLab follows a purely scientific approach combining advanced econometrics methods with up-to-date theory of financial mathematics and data processing technology. The engine behind the platform is based on advanced econometric methods developed by the team of CromoLab. The team consists of professors in econometrics, mathematical finance and information technology (IT), supported by PhDs and researchers in econometrics, machine learning, financial mathematics as well as IT. From econometric point of view, the technology that we develop is based on a selection of papers published by members of our team and their co-authors in the last few years. The main building blocks of the statistical engine driving behind the platform stem from (Van Dijk et al. 2013), (Johansen and Gatarek 2017), (Ardia et al. 2016) and (Ardia, Hoogerheide, and Gatarek 2017). This list is definitely not exhaustive. We explore years of academic and professional experience of CromoLab team members. Due to combinations of state-of-the-art techniques from multiple disciplines of the contemporary econometrics, the CromoLab platform constitutes a coherent and complete investment analysis tool for cryptocurrency market. It covers entire chain of investment decision making process: market price trend analysis, signal processing, portfolio construction, risk analysis. The most distinctive feature of this project is its innovative character in terms of combining two spuriously opponent fields of econometrics into one coherent mechanism. By that We mean mixing classical econometrics philosophy on one side, and Bayesian econometric paradigm of computational, simulation-based, inference. Bringing together classical, large sample based statistical inference, and implementational beauty of Bayesian simulation based approach, positions this project as one of the most innovative project on the ledger of modern econometric inference. It allows for building a solid platform which covers plethora of modern analytical techniques to deliver the best possible insight around portfolio analysis in cryptocurrency market. The main components of the platform are: **trend signalling tool**, **portfolio constrution tool** and **market making tool**. The trend signalling tool identifies the current trend in the market (rising or declining trend) for each cryptocurrency, so that the investor can decide on the direction of her market exposure, long or short in the underlying cryptocurrency. Portfolio construction tool is designed to hedge the investment in the trending cryptocurrency with oppsite exposures in other cryptocurrencies, statistically related to the underlying to provide the investor with variance minimizing portfolio and reduce the market risk to the minimum. Finally the market making tool is though for investors, which have ambition to trade at high frequency. The accurate statistical algorithms explore the relation of iiliquid cryptocurrecies to more liquid ones to determine the accurate market price. Based on this market making tools every investor can set the price in the illiquid cryptocurrencies with extreme accuracy.

Contents

1 Introduction

Bitcoin *puts a question mark on the fractional banking model we know today*- Christine Lagarde, the Managing Director of the International Monetary Fund said in a remarkably frank talk at a Bank of England conference. She assumes that the cryptocurrency can displace conventional banking and has unquestioned potential to challenge the monopoly of national monies as payment mean. Apart from direct impact on worldwide economy and settlement systems, it can have a giant influence on financial industry itself. A huge part of financial institutions, hedge funds, investment banks and high net worth private investors that are currently active in the foreign exchange market might observe a rapid decrease of market activity and thus declining returns.

Nowadays the investment industry makes use of plethora of analytical tools to analyze the foreign exchange markets. However, because of the rapid changes in the market and increasing exposure to the cryptocurrency the standard analytical tools become obsolete. There is a scarcity of tools which provide proper signaling and portfolio construction methods for cryptocurrency market. For the time being they hardly exist. CromoLab sets an objective to develop such technology. It brings together modern econometric knowhow and melts it with the most modern information technology to create a very flexible cryptocurrency investment platform based purely on analytical methods. The team of CromoLab has ambition to create technology centered around machine intelligence based on most advanced methods of computational statistics.

CromoLab follows a purely scientific approach combining advanced econometrics methods with up-to-date theory of financial mathematics and data processing technology. The engine behind the platform is based on advanced econometric methods developed by the team of CromoLab. The team consists of world class professors in econometrics, mathematical finance and information technology (IT), supported by PhDs and researchers in econometrics, machine learning, financial mathematics as well as IT. From econometric point of view, the technology that we develop is based on a selection of papers published by the members of our team or their co-authors in the last few years. The main building blocks of the statistical engine driving behind the platform stem from (Van Dijk et al. 2013), (Johansen and Gatarek 2017), (Ardia et al. 2016) and (Ardia, Hoogerheide, and Gatarek 2017). This list is definitely not exhaustive. We explore years of academic and professional experience of CromoLab team members. Due to combinations of state-of-the-art techniques from multiple disciplines of the contemporary econometrics, the CromoLab platform constitutes a coherent and complete investment analysis tool for cryptocurrency market. It covers entire chain of investment decision making process: market price trend analysis, signal processing, portfolio construction, risk analysis.

This white paper presents the methodological underpinnings of econometric methods behind the technology that CromoLab is creating. It presents the functionalities of the technology and gives some understanding about services, products, that the technology offers to the investment industry. For more accessible information on business model please consult our position paper. Despite of some operational elements, the mail objective of this paper is to present the mathematical foundations of our development.

2 Technical introduction

Recording of asset price over certain time forms into a sequence of measurements called formally time series. The analysis of time series dates back to 1940s and it has remained a well established discipline of econometrics henceforth. Definitely one of the most distinguished works in this stream is a paper of Prof. Johansen, (Johansen 1988), which has defined the discipline of economic time series analysis for the next 30 years. (Johansen 1988) develops methodolgy to test the exact form of dependency across a set of time series associated to a group of economic variables. In a nutshell this philosophy assumes that there exists a common trend, or a group of common trends, which drives the group of time series. The methodology developed by Prof. Johansen makes it possible to identify those trends. From statistical point of view those trends form time series as well. They can be measured upon measurements of underlying economic variables in the analyzed system. The subdiscipline of time series analysis designed for identification of common trends among many time series is called cointegration analysis. CromoLab platform is built around analytical solutions that identify the common trends driving the cryptocurrency prices in spirit of modern simulation-based approach to cointegration analysis. That allows not only for proper trend identification but moreover for portfolio construction and market making. The theoretical foundations of CromoLab has been developed in (Johansen and Gatarek 2017) and are highly inspired by advanced treatment presented by (Johansen 2006).

In the early papers of 1980s and 1990s, cointegration analysis has mostly been applied to macroeconomic time series. They usually describe main economic indicators in a macroscale. With deeper and deeper understanding of financial markets dynamics, cointegration analysis started to play substantial role in modeling of asset prices and their linkages. That was somehow natural as asset prices are modeled with random walks, which from the theoretical point of view define time series processes underlying cointegration analysis.

Random walk is a simple yet very general process that ensembles the idea of randomness over the time. In a nutshell a random walk should be perceived as a process which, starting at some level, can grow or decrease by one unit every period. If the process is symmetric the probability of up and down movement are equal. In case of asymmetric random walk the probability of increase differs from the probability of downward movement. Both are defined by a proper probability distribution associating a variable $p \in (0,1)$ with a likelihood of the process to increase by one unit. Then, $1-p$ automatically stands for the probability of decrease. Typically, the mathematical theory of random walk assumes that those probabilities equal 0.5 for both up- and downmove. According to the CromoLab view, in applications to financial markets those probabilities should definitely not be assumed equal. That is an aftermath of alternating trends which occur in the asset price. CromoLab associates the trends in the cryptocurrency price with fluctuation of this probability measure. The probabilities fluctuate, what results in asset price trending immediately. Why? The probability $p > 0.5$ makes generation of $+1$s more probable than generation of -1. Then, the rising trend is more likely to occurr than the declining trend. Such an interpretation of the market price trending brings it down to estimating the probability distribution driving the random walk behind the prices. The Momentum strategy of trend following is a direct consequence of time-varying probabilities of ups/downs. The probability level itself should be interpreted as a determinant of a current market cycle over rising and declining trends. The estimation of the fluctuations of random walk probability distribution $(p, 1-p)$ over time is one of the key components of the statistical engine behind the CromoLab platform. Another building block is the cointegration analysis itself.

Why is cointegration analysis so important in terms of price series in cryptocurrency market? First of all it allows for modeling and relating one cryptocurrency price time series to other cryptocurrencies. Secondly, based on methodology recently developed in (Johansen and Gatarek 2017) we can build portfolios of cointegrated assets under minimal variance principle or maximal Sharpe ratio (defined as expected return over risk). The principles developed in (Johansen and Gatarek 2017) form optimal hedging portfolio. Further, as cointegration assumes a random walk nature of asset price, but does not impose any other conditions, it is definitely applicable to the situation where the probability distribution of random walk evolves over time and is not fixed at equal probabilities $(0.5, 0.5)$. To that end it needs to be combined with proper estimation methods.

Combining detailed analysis of cointegration with accurate time specific probability distribution for random walk up- and down-movements constitutes a complete methodology for optimal portfolio construction and hedging in platform proposed by CromoLab. The estimates of probability give indication on the direction of investment to enter. Given the sign of market position, investment, cointegration analysis allows proper portfolio construction.

3 Applicability of econometrics to blockchain analytics

The question arises around the applicability of traditional analysis of financial markets to the new economy of cryptocurrencies. Indeed, the nature of price formation might be very different in the cryptocurrency market due to decentralization of blockchain system that stands behind the cryptocurrencies. There is no central bank for money supply as in case of traditional currencies active in foreign exchange market. However, based on research performed by CromoLab in terms of statistical properties, nothing differs substantially from traditionally measured financial asset prices. Cryptocurrency price process can be modeled with random walk and, as a consequence, cointegration can be applied. We refer to research of (Grassi and Catania 2017), who have researched statistical properties of cryptocurrency price series. To sum up, substantial part of time series models that have been developed in the financial econometrics so far, can be transplanted into analysis of cryptocurrencies.

The use of cointegration for analyzing financial data is well established over the last 20 years. Regarding the most influential papers in the discipline, the problem of price discovery is discussed by (Hasbrouck 1988), (Lehmann 2002), (Jong and Schotman 2010), and (Grammig and Schlag 2005). (Gatev, Goetzmann, and Rouwenhorst 2006) study pairs trading, and continuous time models with a heteroscedastic error process are developed by (Duan and Pliska 2004). (Alexander and Dimitriu 2005a), and more recently (Juhl, Kawaller, and Koch 2012), studied optimal hedging using cointegration. (Ardia et al. 2016) have presented how to apply a specific restriction on cointegration model to make it fully applicable to financial applications. Finally (Johansen and Gatarek 2017) have developed a methodology for optimal portfolio construction based on asset prices driven by random walks with portfolio weights depending on the hedging horizon.

4 Theoretical foundations of CromoLab statistical engine

To sum up the discussion so far, there are a few key building blocks of the statistical engine behind the platform.

First of all we assume that the cryptocurrency rate as any other exchange rate or financial asset price can be modeled with a random walk.

Secondly we assume that there is a probability distribution that stands behind this random walk, in terms of up- and downmovement. Then, it is assumed that the evolution of this probability over time can lead to periods of rising trends and downturns.

Furthermore, we assume that the historical quotations of the cryptocurrency rate can be explored for estimating on the relation between the price movements and the probability distribution governing the random walk behind.

Finally we assume that the entire spectrum of price series which are representing the cryptocurrencies can be modeled by means of cointegration analysis and if that is the case, we have the entire spectrum of methods developed by econometrics available for portfolio construction in such a market environment.

5 Products

The statistical engine which drives in the background has one objective. It delivers analytics which is displayed by the CormoLab platform. The analytical platform is the main product developed by CromoLab team. However the customers would have options to buy insight at different level of detail and accuracy. In what follows we present the products to be offerred by the CromoLab platform. Diagrammatic representation of the platform together with corresponding engine components is depicted in Figure 1.

RISING / DECLINING TREND PROBABILISTIC SIGNAL & EXPECTED TREND REVERSAL TIME Based upon accurate probability model estimated for each cryptocurrency, the customer obtains signal which present current market sentiment, upward or downward trend, with information on its lenght up to date and its expected duration. Initial analysis has shown that trends are definitely alternating. Downward movements are followed by upward trends and vice versa. Recurrent character of these events makes it possible to model it by means of trigonometric series. CromoLab has researched those cycles based on methods developed in (Van Dijk, Harvey, and Trimbur 2007). These techniques have been combined with (Van Dijk and Kleijn 2006) into complex methodology applicable for CromoLab prediction challenge. The system is able to deliver the expected time of arrival at the equilibrium which is interpreted as the end of the currently observed trend (cycle) and, potentially, beginning of the contrary trend. The equilibrium is interpreted as a phase when market has no direction or is just after the end of a trend. Such analysis allows for momentum trading based on the identified trend and presents a possibility for trend following.

OPTIMAL PORTFOLIO Apart from the trend identification functionality, which constitutes signalling component of the statistical engine, CromoLab platform allows for portfolio construction around a selected cryptocurrency. An investor is supposed to select a trending cryptocurrency to invest in it based on insights delivered by the platform. If the selected cryptocurrency follows an upward trend, the investor wishes to enter a long position in this cryptocurrency. The decision to be made concerns following an outright position, unhedged by a contrarian investment in other cryptocurrency, or, alternatively, she can hedge this position with a proper portfolio around it. To that end the methodology in (Johansen and Gatarek 2017) is applied. The fact that the cryptocurrencies can be modeled by means of random walks, which in terms of financial markets are definitely driven by common trends, and the fact that they are characterized by extensive amount of correlation among each other, opens a wide application potential for methods developed in (Johansen and Gatarek 2017). The methodology assumes that the assets can be modeled with a cointegration model. Given the selected cryptocurrency traded by the investor, the system selects the optimal set of variables to enter in the cointegration model. This set is selected among all the other cryptocurrencies analyzed by the platform by means of proper classifications algorithms which

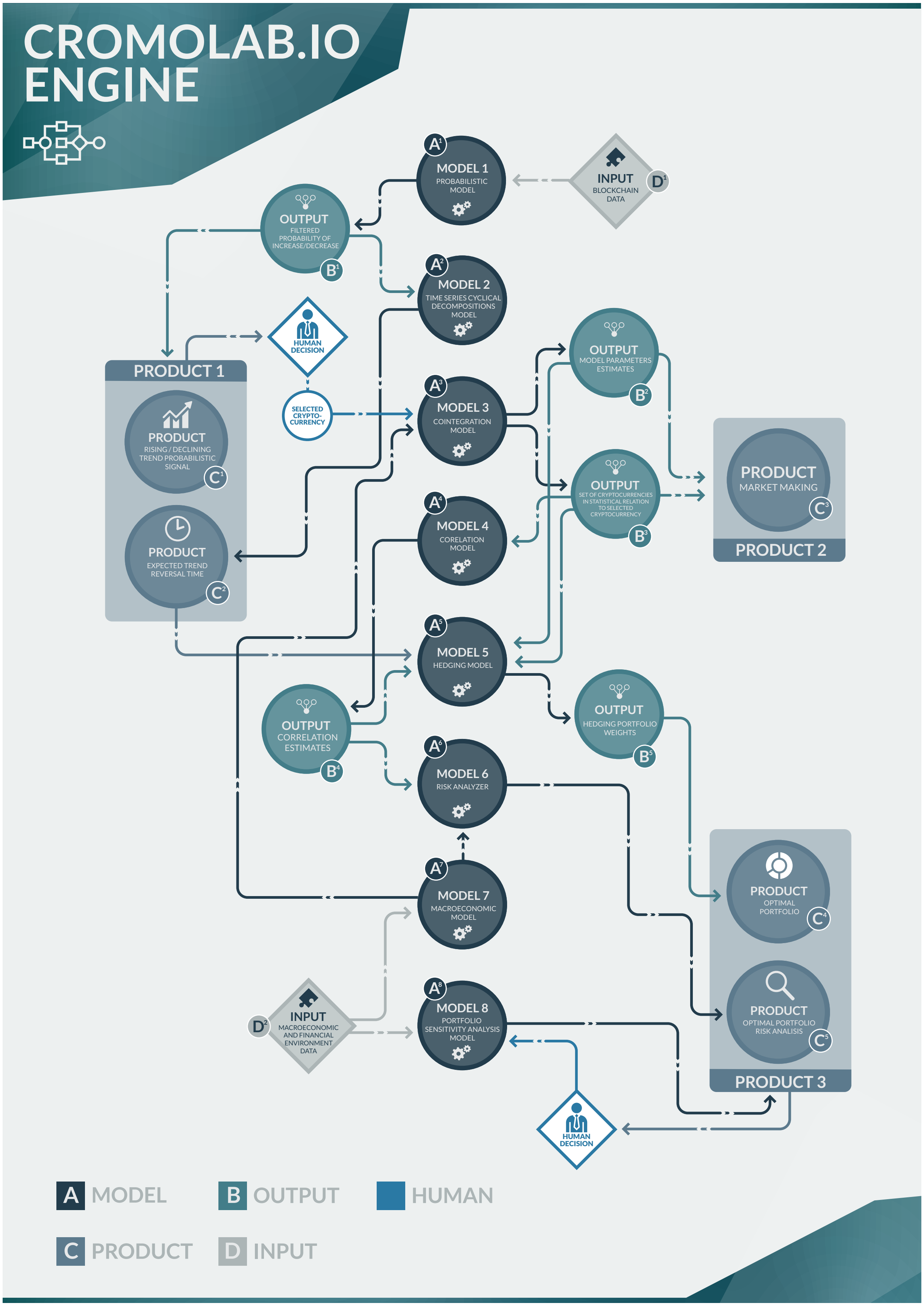
CROMOLAB.IO ENGINE
A1 MODEL 1 PROBABILISTIC MODEL
INPUT BLOCKCHAIN DATA D1
OUTPUT FILTERED PROBABILITY OF INCREASE/DECREASE B1
A2 MODEL 2 TIME SERIES CYCLICAL DECOMPOSITIONS MODEL
HUMAN DECISION
SELECTED CRYPTO-CURRENCY
PRODUCT 1
PRODUCT RISING / DECLINING TREND PROBABILISTIC SIGNAL C1
PRODUCT EXPECTED TREND REVERSAL TIME C2
A3 MODEL 3 COINTEGRATION MODEL
OUTPUT MODEL PARAMETERS ESTIMATES B2
OUTPUT SET OF CRYPTOCURRENCIES IN STATISTICAL RELATION TO SELECTED CRYPTOCURRENCY B3
PRODUCT MARKET MAKING C3
PRODUCT 2
A4 MODEL 4 CORELATION MODEL
A5 MODEL 5 HEDGING MODEL
OUTPUT HEDGING PORTFOLIO WEIGHTS B5
OUTPUT CORRELATION ESTIMATES B4
A6 MODEL 6 RISK ANALYZER
A7 MODEL 7 MACROECONOMIC MODEL
PRODUCT OPTIMAL PORTFOLIO C4
PRODUCT OPTIMAL PORTFOLIO RISK ANALISIS C5
PRODUCT 3
D2 INPUT MACROECONOMIC AND FINANCIAL ENVIRONMENT DATA
A8 MODEL 8 PORTFOLIO SENSITIVITY ANALYSIS MODEL
HUMAN DECISION
A MODEL
B OUTPUT
HUMAN
C PRODUCT
D INPUT

Figure 1: CromoLab platform: products and statistical engine

divide the spectrum of available cryptocurrencies into applicable and nonapplicable as hedging counterparties for a particular cryptocurrency. In fact the statistical engine estimates variety of models in the background and selects most appealing specification from the statistical point of view. Based on this specification, the methodology developed in (Johansen and Gatarek 2017) is applied to deliver the optimal portfolio weights leading to minimum portfolio variance in a given time horizon. This horizon is derived as an expected portfolio holding periods and it is indicated by an accurate model based on the time series properties of probability which has been discussed above and is estimated based on (Van Dijk, Harvey, and Trimbur 2007) and (Van Dijk and Kleijn 2006). The initial research performed by CromoLab indicates that the first derivative of the probability curve indicates highly cyclic behaviour and its dynamics is able to time the trend reversals accurately. Naturally, despite of an automatic horizon length selection, an investor can define the horizon in an ad hoc fashion and select corresponding portfolio instead of the portfolio inidicated by the cyclical model.

MARKET MAKING Market making, similarly to hedging, is based on cointegration model. In that case the cointegration model is applied to calculate the cryptocurrency price based on other cryptocurrencies and given the statistical relation between them. The statistical relation is based on cointegration model and it specifies the link between the underlying cryptocurrency and other cryptocurrencies. The idea is based upon the definition of cointegration model which identifies a long term relation between cryptocurrencies. If such a long term relation between cryptocurrencies exists then the system would revert to it even if some temporary deviations are observed. Any deviation from the long term relation implies some disequilibrium which is a gap that should be closed sooner rather than later by the market forces. Thus by taking a position in one cryptocurrency (whose price deviates from the equilibrium), one can exert a profit. For instance, the investor might wish to open a position in Zcash, which seems to be out of equilibrium in an analyzed period of time, but does not know the fair value of this cryptocurrency. This market as any other tends to under and overestimate the value of the cryptocurrencies in some periods of time. Due to high volatility, it is even more frequently observed than in case of more traditional markets. They go back to equilibrium quite fast, but if one knew when the price remains in disequilibrium, then the profit taking opportunity would be substantial. Imagine that according to the cointegration model, the Zcash is in relation to Bitcoin, Ethereum and Dash. If this relation is strong and implied by a stable cointegration model, we can identify the current fair (model implied) price of Zcash based on the current prices of Bitcoin, Ethereum and Dash. If, for some reasons, the Zcash market price deviates from the model implied price, there is possibility for profit taking. This technique is called market making and it is typically applied by institutional investors, mostly hedge funds, in particular in illiquid assets. CromoLab offers technology for market making in cryptocurrency market.

6 Data science market environment and CromoLab target group

CromoLab has grown with the broader trend of increasing interest in analytical services often referred to as data science. Data science, and its more IT wing - big data, represent how technology has evolved past simple functions like spreadsheet functions or word processing to harvesting insights and analytics from the (often massive) amount of data collected by organizations. This data can be based upon almost literally anything; consumer spending habits, theater box office records, meteorological activity, sunspot behavior; anything that can be tracked and from which useful insights can be gleaned. Cryptocurrency market is another field of applications of data science, with one reservations: it is hardly possible to analyze cryptocurrency in another way as with rigorous statistical techniques. Blockchain is just a massive data lake to analyze it differently. No pure fundamental analysis would ever work with blockchain.

Data science isn't new, but its popularity has skyrocketed as of late, in part because of technological advances permitting the storage and processing of large amounts of information, as well as ever-more competitive marketing demands to build best of breed organizations.

Regarding the customers of CromoLab platform, the products delivered by CromoLab shall be classified as data-scientific service. Analytical reports meet broader and broader interest both in financial industry and beyond. Big data has definitely become a part of modern business model in every industry. Data scientist is probably the hottest job of 21st century so far. Quantitative analytics became an industry on its own in finance and it brings in an impressive amount of money all over the world. In 2015 alone professional traders spent over $50 billion on purchasing financial market data. 4 billion of it was spent on professional analytical services and systems, based on predictive analytics for algorithmic trading as well as portfolio optimization in wealth management context. After 100 years of theoretical development, statistics seem to become as important as business strategy. By 2020 the amount spent for analytics will increase approximately six times, according to multiple sources. The B2C financial information market for non-professionals is huge. 54% of US residents have bought shares at least once in their lives, in China about 30% of residents are actively engaged in stock trading and moving to alternative form of investment as well. This trend is somehow natural as an overwhelming information makes it almost impossible to read and analyze everything published on a specific asset. Therefore CromoLab team bets for simplicity. CromoLab associates the top experts in data science, information management and quantitative finance to build complex platform which provides extremely accessible analytical service for cryptocurrency market participants. CromoLab platform products are highly complicated inside but are presented to the user in simple and accessible way.

Currently the financial market observes a gap in terms of cryptocurrency-specific analytics. The demand is growing tremendously due to rapid development of cryptocurrency market and progressing tokenization of services on-line. The novel character of this phenomena is not accompanied by sufficient research and know-how. The reason for this gap partly derives from emerging character of the cryptocurrency market. However to substantial extent the scarcity of analytical service roots from lack of blockchain data scientific skills in the job market. There are experts which can analyze large data sets with big data tools. At the other side there are experts which can build complex statistical mode. The cryptocurrency price prediction equation requires top experts in both of this fields, working together. CromoLab has managed to union such experts. Both top statisiticians as well as top data processing experts work for us.

7 Why cryptocurrency can not go around without data analytics?

Cryptocurrency market is different from any other financial market before. The price of the cryptocurrency is not implied by the supply managed by a central bank. It is neither a result of interest rate policy, purchasing power parity nor anything related to old, well known, macroeconomic inference. It is interesting that the concept of interest rate does not actually exist in case of a cryptocurrency. It is all about transactioning. The perception of value of cryptocurrency is what ultimately gives it value, what people are willing to put in to get a unit of cryptocurrency, be it time, fait money or labour.

The cryptocurrency value fully relies on digital foundations. For the first time in the history, the statistical skills have indisputable advantage over the fundamental analysis of an asset. For last three decades quants and fundamental analysts has been in continuous dispute over who is right in approach to market analysis. In case of cryptocurrency this dispute is pointless. It is impossible to actually analyze cryptocurrency whithout analyzing the blockchain dynamics... and this is plenty of data, so its about statistics. Analysis of an asset which is fully digital and backed by information flow can only be performed based on statistical technique. CromoLab seems to be one of the first, if not the first undertaking, of rigorous treatment of cryptocurrency price analytics based on statistical analysis of blockchain. Inference based on macroeconomic principles does not work in case of blockcahin. The volatility of cryptocurrency can shoot in the sky with all the economic indicator remaining stable. (Grassi and Catania 2017) have performed in depth statistical analysis of cryptocurrenc prices. They found that in case of cryptocurrencies, on average, volatility increases more after negative shock than after positive shock as in the equity market, hence, cryptocurrency time–series incorporate the so–called leverage effect with some degree of heterogeneity across the series. Of course, there are some fundamental factors impacting value of cryptocurrency. For instance, in case of Bitcoin, it is usable for payments on a reasonably high and ever increasing scale, meaning that its utility is high. Its high difficulty and energy usage gives it a reasonably high price and as such can be used for an investment. The changes to utility can cause price volatility. In the case of Ether, as it was designed a smart contract platform this is a practical utility, which increased the price of Ether over many other alternative cryptocurrencies. Despite of those fundamental factors, the amount of complexity of those systems makes it almost impossible for the standard fundamental reasoning rules to analyze trends properly. Because of that, the data scientific approach appeals as particularly promising in blockchain analysis.

Irrespective if it is novel character of the market or scarcity of proper data scientific skills what causes the burden, the analytical service currently available in the cryptocurrency market is extremely scarce. This is a very specific market where deep data scientific skills are required to analyze the observed price processes. The blockchain data available for analysis is much more involved subject of analysis than traditionally observed data of stocks, foreign exchange or bonds. Those markets are definitely matured and can be analyzed with thousands of platforms which came to existent over last twenty years. Cryptocurrency are currently analyzed only by individuals who have skills to research the blockchain data properly. Massive character of this information requires scarce skills.

CromoLab sets an ambitious objective to bring such an analytical platform to a wide audience. The volatility observed in the cryptocurrency market and related return-on-investment potential is extreme, as in every emerging markets in the history of financial economics. CromoLab wishes to give the individual investors as well as institutions a possibility to participate in this market in an informed way, based on concrete outcome of analytical data processing.

CromoLab platform designed and to be implemented by the CromoLab is supposed to be a first analytical framework which provides full spectrum of analytical service for the cryptocurrency market. Starting with trend signal processing, going over portfolio construction and finally market-making, CromoLab platform covers all needs of informed cryptocurrency investor.

8 Motivation: fluctuations of the random walk

8.1 How random is the random walk?

The basic idea underlying the cryptocurrency trend signaling, which constitutes the main component of the statistical engine to be developed, is derived from the theory of random walk fluctuations. This theory has been researched in the middle of twentieth century thanks to (Lévy 1939) and has remained a very appealing discipline of modern statistics. In probability theory, the fluctuations of random walk are associated with arcus sine laws. Those are collection of results for one-dimensional random walks which present unituitive rules governing evolution of random processe. For introduction to the topic the reader is refferred to (Feller 1957).

The results concerning fluctuations in coin tossing show that widely held beliefs about the law of large numbers are fallacious in terms of random walk. The implications of random walk fluctuations are so amazing and so at the variance with common intuition that even sophisticated professors of statistics have doubted that coins, which constitute the simples model of random walk, actually misbehave to what would be expected from the random walk based upon common intuition.

The theory developed in (Lévy 1939) is representative of a fairly general situation that can be described in form of a random path in two dimensional space of time and value (of cryptocurrency price in tis case). In this chapter we present the necessary information on the theory of random walk fluctuations for understanding the main idea of methodology we apply in CromoLab engine. We are motivated principally by the unexpected discovery that this theory can be treated by elementary method of mathematics, in particular basic combinatorics and fairly basic statistical theory.

We start with defining the rules of typical coin-tossing game and its relation to the simple random walk. Let's start with an assumption that the initial fortune equals to zero at the beginning of the game. Every period (it can be a second, day, month etc. depending on the selected data granularity) we toss a fair coin. Based on its utcome, each time we either collect 1 unit of value (can be 1 usd for instance) from our opponent or we pay it out. Automatically the process of random walk which models the game increases or decreases by one unit in a given period. The line representing the evolution of wealth generated by the player in such a game is a perfect example of random walk evolution over the time. In Figure 2 we present a simple example of random walk evolution over time. In this case the random walk evolves over 12 periods which might be interpreted as 12 games played between two players as skizzed above.

As the number of games played increases, by the Law of Large Numbers, the fraction of games with positive outcome approaches 1/2 (and negative as well) for a symmetric coin. In the course of the game, however, fortune of a player is likely to fluctuate, changing sign from positive to negative and back. We pose a question: after a large number of n games have been played, what is the fraction of time when the total wealth remained in the winning zone? This question is a key problem of random walk fluctuations theory. In what follows we present some components of this theory which aim at answering this question.

By obvious symmetry, in the time between the consequitive draws, when the total wealth becomes zero, the fortune is equally likely to stay in the positive and negative area. As more games are played, more draws will occur, which seem to imply that the fraction of time spent in positive area goes to 1/2. This would be standard intuition put forward by everyman asked the above mentioned question. In fact this intuition is completely wrong! The fraction of time spent in positive area is more likely to be close to extremities 0 or 1 than to be close to 1/2. The results are startling. As

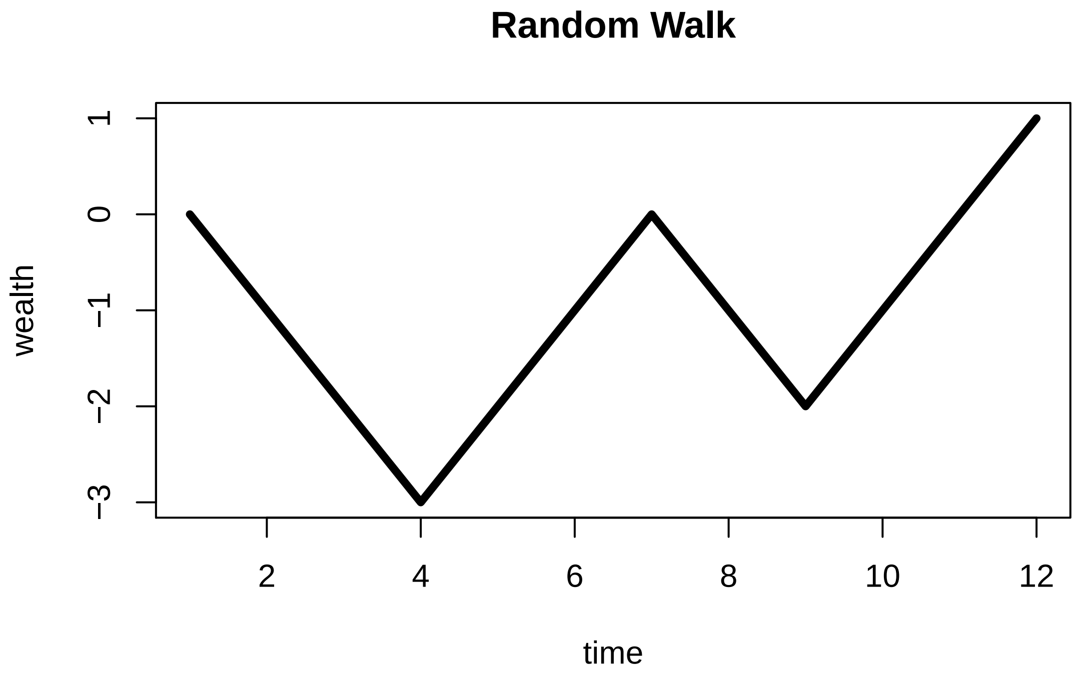

Figure 2: Example of simple random walk.

mentioned above, according to the widespread belief a so-called law of large numbers should ensure that in a long coin-tossing game each player will be on the winning and losing side for about half the time, and that the lead will pass not infrequently from one player to another one. Lets imagine then a huge sample of records of ideal coin-tossing games, each consisting of $2n$ trials. (It is $2n$ rather than n as the draw can only happen in even periods.) We randomly pick one such a period and identify the period of last draw, where the number of accumulated heads and tails were equal. This number is even, and we denote it by $2k$ (so that $0 < k < n$). Frequent changes of lead would imply that k is likely to be relaitvely close to n, but in practice this does not need to be so. In fact, the distribution of k is symmetric (any value of k has exactly the same probability as the value of $n - k$). Furthermore, the probabilities near the end points, so near $2n$ are the greatest. The most possible values for k are the extremes of 0 and n. The intuition lead to an erroneous picture of probable effects of coin fluctuations. Figure 3 presents profile of such probability for each k, where $n = 4$, thus $2n = 8$.

The understand the probability profile presented in Figure 3 i.e. the relation of time spent over zero axis to the total length of the random walk curve, we need to take a closer look at the nature of the problem. Let us assume that we work with the simplest possible definition of a random walk, which is defined as a process which starts at period 0 with value 0 and spans over $2n$ periods, and each period it can increase or decrease in value by 1 unit with probability 1/2 both. From a combinatorial point of view we shall be concerned with arrangment of $2n$ plus ones and minus ones. If we refer to $2n$ as the length of the random walk path, then there are 2^{2n} different possible paths which can realize with this length of the process. Why? Because each period, called alternatively as an epoch, $k = 1, \ldots, n$, the process have two options to evolve: either $+1$ or -1. Then in two consequtive

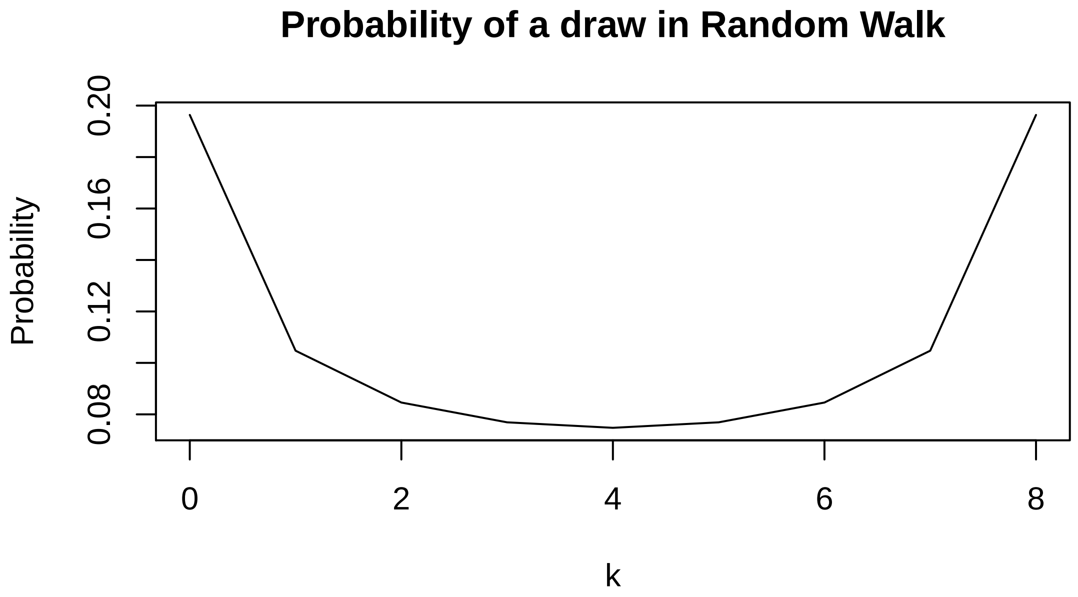

Figure 3: Probability of a draw in a simple random walk with number of periods 2n=8. A draw is represent by a period of time when random walk crosses a zero line.

periods it has 2 × 2 options to evolve, and in $2n$ periods, it can evolve in $\underbrace{2 \times 2 \times 2 \ldots 2}_{\text{2n}}$ different ways. In Figure 4 we present a lattice which represents all the possible paths for the random walk over 6 periods with 3 upward and 3 downward movements. We present also corresponding code in R statistical computing language. For most of the figures presented in this paper associated code is also attached.

```
## the combinatorial approach to random walk construction

# assume some level p for +1s
p <- 3
# assume some level q for -1s
q <- 3

# total number of steps taken (periods) by the random walk, x, is just a
# summation of number of occurrence of digit +1 and -1
n <- p + q
# the terminal level y is just the excess frequency of occurrence of the
# digit 1 over -1 among the x places (periods)
x <- p - q

# the combinations of p out of x places
numCombn <- dim(combn(1:n, p))[2]
combnRWs <- matrix(-1, numCombn, n)
```

```
cumSumCombnRWs <- matrix(NA, numCombn, n)

allCombs <- combn(1:n, p)

for (i in 1:numCombn) {
    combnRWs[i, allCombs[, i]] <- 1
    cumSumCombnRWs[i, ] <- cumsum(combnRWs[i, ])
}

cumSumCombnRWs <- cbind(rep(0, dim(cumSumCombnRWs)[1]), cumSumCombnRWs)

matplot(t(cumSumCombnRWs[, ]), type = "l", lwd = 4, xlab = "time", ylab = "wealth",
    main = "Random Walk paths")
```

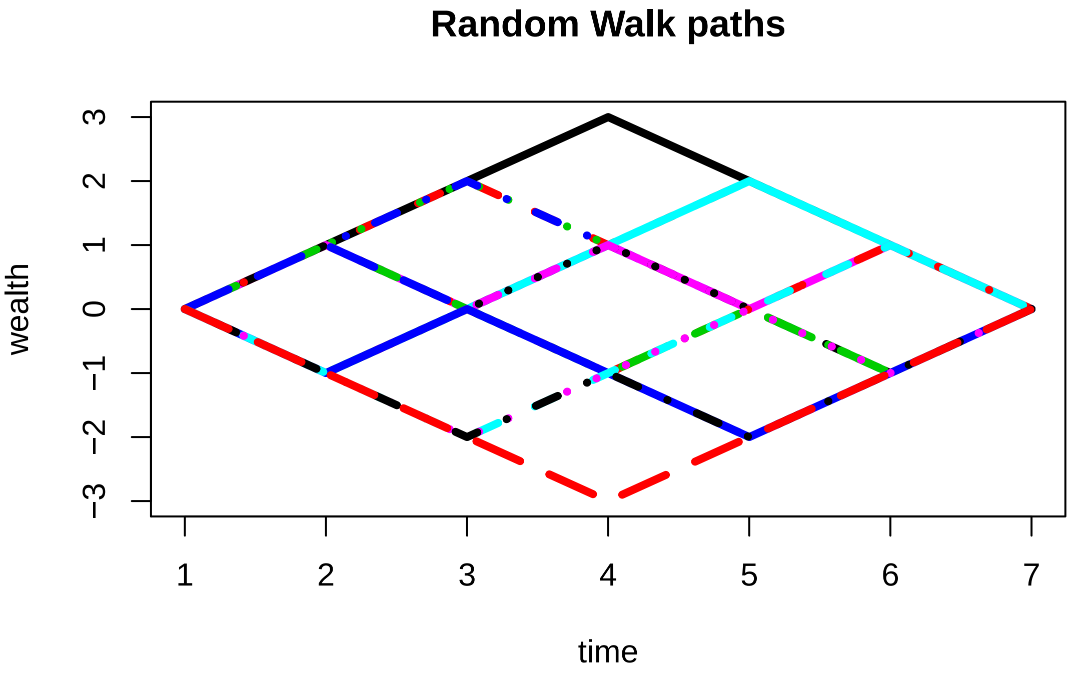

Figure 4: All possible paths for a random walk with 2n = 6.

Irrespective of the trajectory the random walk traverses, the initial wealth and the terminal wealth are equal. This is a consequence of equal number of $+1$s and -1s on each path. If the process rises in 3 out of 6 periods and decreases in remaining 3 periods, then it will always go back to the initial value irrespective of the order of ups and downs.

If we loosen the restriction of equal number of ups and downs and start to generalize the process, the initial and terminal wealth would not be equal. For instance lets imagine that we assume that among the $2n$ epochs we observe p plus ones and q minus ones. Then, naturally $2n = p + q$ and the value at the period $2n$, so the terminal value of all such paths equals $p \times 1 + q \times (-1) = p - q$.

In Figure 5 the situation with $p = 4$ and $q = 6$ is presented and in Figure 6 a reverse order of p and q is applied. In Figure 5 all paths end in terminal wealth of $p - q = 4 - 6 = -2$, whereas in Figure 6 they end in $p - q = 6 - 4 = 2$. We observe that the dominance of number of ups over the number of down movements automatically ends in a terminal value in the positive territory, and vice versa.

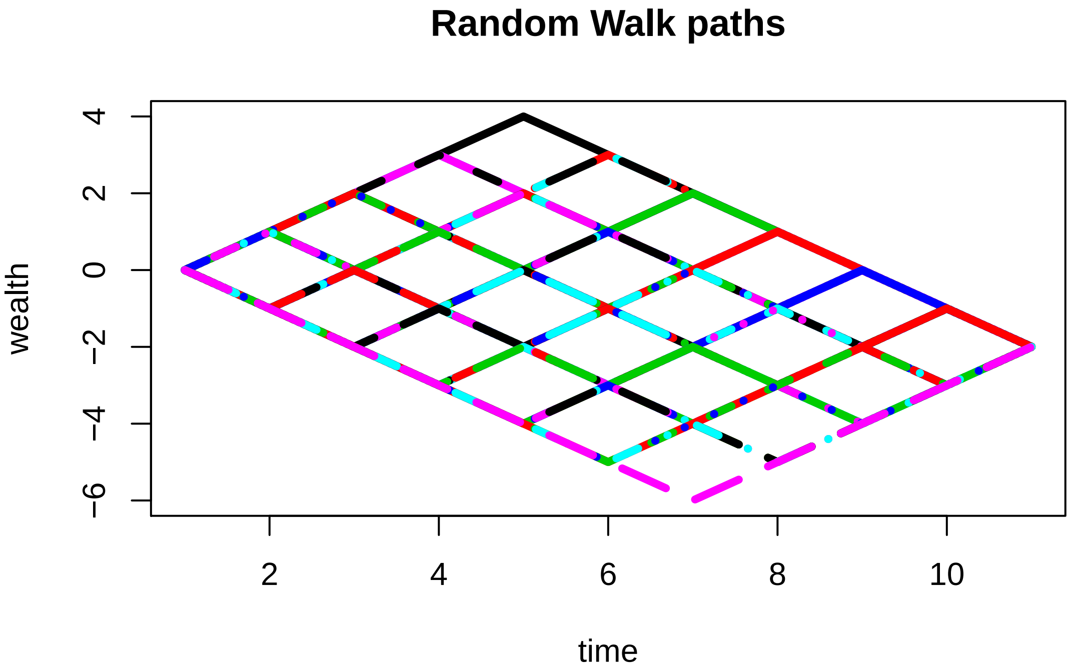

Figure 5: All possible paths for a random walk with 2n = 10 with p=4 and q=6

If we denote $p - q$ with x, then a path from origin to an arbitrary point $(2n, x)$ exists if and only if $2n$ and x remain in the abovementioned relation to p and q i.e.

$$2n = p + q$$
$$x = p - q.$$

Thus each point $(2n, x)$ in the two dimensional space automatically implies the corresponding values of p and q. For instance, for the random walk paths that connect the origin $(0, 0)$ with the point $(6, 2)$, according to the formulas we need to solve the system of simple equations

$$6 = p + q$$
$$2 = p - q$$

which is satisfied by $p = 4$ and $q = 2$. Thus, all the paths which lead the process to the value of 2 in 6 steps result from 4 plus ones and 2 minus ones. How many paths of that sort can be constructed? The answer to that question is fairly simple and stems from the basic combinatorics: that is a number of ways we can distribute 4 balls with a ticker plus ones in 6 bins. Thus the answer is: $\binom{6}{4}$ which can be generalized to

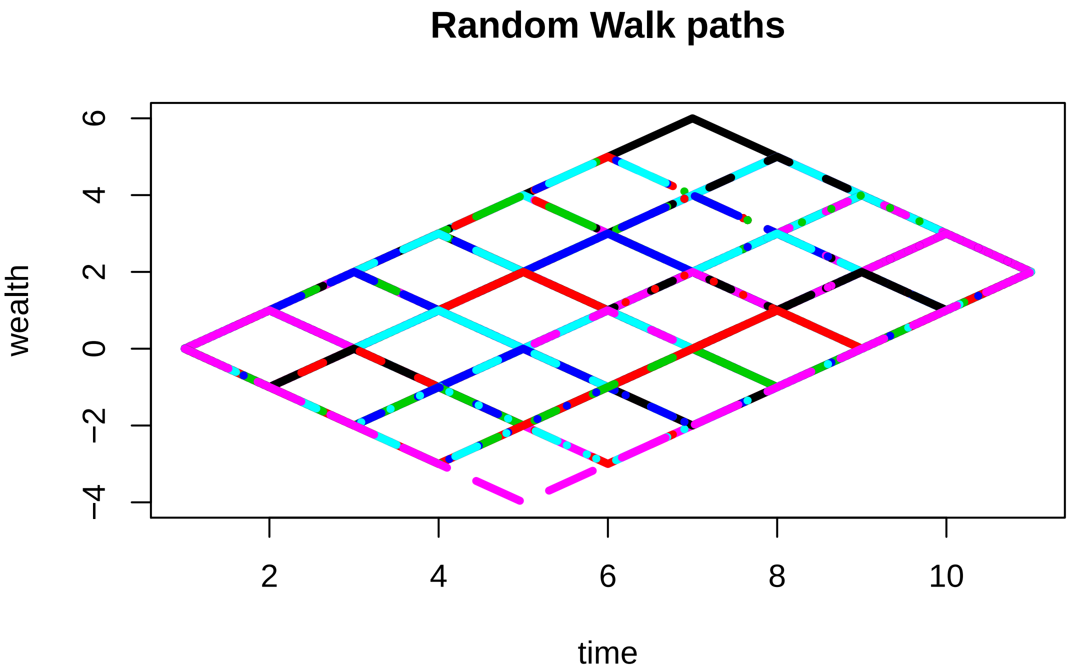

Figure 6: All possible paths for a random walk with 2n = 10 with p=6 and q=4

$$\binom{2n}{p} \tag{1}$$

.

Due to out interest in the amount of time spent over the axis, we are mostly devoted to the paths which revert to a zero at some point of time, so where a draw occurs at some point on the random walk path. Let us denote by u_{2n} the probability of a return to zero at period $2n$

$$u_{2n} = P(S_{2n} = 0),$$

where S denotes a path of random walk (S stems from the fact that we define the random walk as a sum of a sequence of plus and minus ones). Given the amount of paths in Equation 1 we can compute the probability u_{2n} as

$$u_{2n} = \binom{2n}{n} 2^{-2n}. \tag{2}$$

The first term $\binom{2n}{n}$ is motivated by the fact that n out of $2n$ plus ones must occur on the path so that it reverts back to 0 in $2n$ periods. That means that $p = n$ in Equation 1. The term 2^{2n} refers to the total number of all the possible paths between 0 and $2n$, what has been explained above. The important theorem that we base upon is well known as an arc sine law in the theory of random walk:

Table 1: Discrete Arc Sine Distribution of order 8

k=0	k=1	k=2	k=3	k=4
k=8	k=7	k=6	k=5	
0.1964	0.1047	0.0846	0.0769	0.0748

The probability that up to and including period $2n$ the random walk visits the positive area for $2k$ periods ($0 <= 2k <= 2n$) and the negative one for $2n - 2k$ periods is given by

$$p_{2k,2n} = u_{2k}u_{2n-2k}. \tag{3}$$

For a detailed proof we refer for instance to (Feller 1957).

Making use of definition in Equation 2 we can compute the precise expression for $p_{2k,2n}$

$$p_{2k,2n} = \binom{2k}{k}2^{-2k}\binom{2(n-k)}{n-k}2^{-2(n-k)} = \binom{2k}{k}\binom{2(n-k)}{n-k}2^{-2n}. \tag{4}$$

It is easy to check that the numbers obtained for different k add to unity. Therefore $p_{2k,2n}$ constitutes a probability distribution. The distribution that attaches a weight $p_{2k,2n}$ to the point $2k$ is called the discrete arc sine distribution of order n. An example of such a distribution has been presented in Table 1 for $n = 8$, and has previously been displayed in Figure 3.

As we can see the central term has the smallest probability. That reflects the main idea of this theorem. Namely the intuition that that both sides of the axis are visited by the random walk with exactly the same frequency is a completely wrong one. The opposite is true: such a scenario has the lowest probability. Table 1 presents the probability for all possible k's from 0 to $n = 8$. The value associated with a given k represents the probability that the random walk visits the positive area (area above the zero axis) for $\frac{2k}{2n} = \frac{k}{n}$ fraction of periods. For the analyzed case of $n = 8$, the probability that $k = 0$ periods are visited on the positive side is equal to 0.1964. The probability is symmetric: $p_{2k,2n} = p_{2n-2k,2n}$. For instance, the probability for 2 out of 16 periods to be spent in the positive area is equal to the probability that 14 out of 16 periods are spent there. This is a consequence of of random walk symmetry. Such a probability equals 0.1047 according to the Table 1. The fact that the intuition fails completely in case of random walk (coin tossing procedure) is striking. Looking at those probabilities we can observe that in terms of $n = 8$, so 16 periods the probability of not being at all, or being all the time in positve area is $0.1964/0.0784 = 2.6256$ times higher than observing a situation where the same amount of time is spent in both positive and negative areas. Figure 7 presents such ratios for different values of n. This plot has been generated with the code belowe. Each line in this figure corresponds to one value of n, running from 2 (green line) to 40 (red line). We can see that the extreme values of $k/n = 0$ and $k/n = 1$ are substantially more probable to realize than the intuitive scenario of both negative and positive territory being visited with equal frequency of $k/n = 0.5$.

```
for (n in 2:40) {

    k <- 0:n
    df <- data.frame(k, round(choose(2 * k, k) * choose(2 * (n - k), n - k) *
        2^{
            -2 * n
        }, 4))

    if (n == 2) {
        plot(k/n, df[, 2]/df[(dim(df)[1] + 1)/2, 2], type = "l", xlab = "Fraction: k/n",
            ylab = "Probability", ylim = c(1, 6), col = "green")
    } else {
        lines(k/n, df[, 2]/df[(dim(df)[1] + 1)/2, 2], ylab = "Probability")
    }
}
if (n == 40) {
    lines(k/n, df[, 2]/df[(dim(df)[1] + 1)/2, 2], ylab = "Probability", col = "red")
}
```

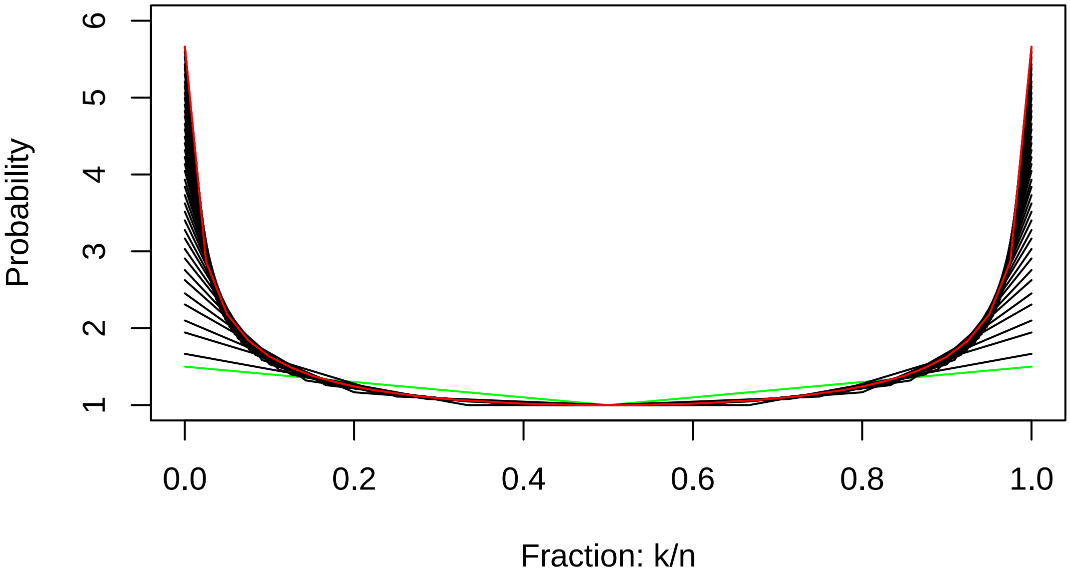

Figure 7: Probability of a draw for different n

To present the full spectrum of results associated with the arc sine distribution we need to express the binomial coefficient in terms of factorials, according to Stirling's formula, see for instance (Feller 1957). It can be shown that

$$u_{2n} = \frac{1}{\sqrt{pn}}.$$

Applying Stirling's formula to Equation 4 we obtain

$$p_{2k,2n} = \frac{1}{\pi\sqrt(k)\sqrt(n-k)}. \tag{5}$$

The latter one is in fact equivalent to

$$p_{2k,2n} = \frac{1}{n\pi\sqrt(k/n)\sqrt((n-k)/n)}. \tag{6}$$

If we denote $x_k = \frac{k}{n}$ we can evaluate the last equation as

$$p_{2k,2n} = \frac{1}{n} f(x_k), \tag{7}$$

with $f(x) = \frac{1}{\pi\sqrt(x(1-x))}$. This function results in a shape presented in Figure 8 with its height varying depending of n. This function is referred to as arc sine probability density function.

```
fraction <- (1:100)/100
plot(fraction, 1/(pi * sqrt(fraction * (1 - fraction))), type = "l", xlab = "Fraction: k/n",
    ylab = "Probability", main = "Arc sine Pdf")
```

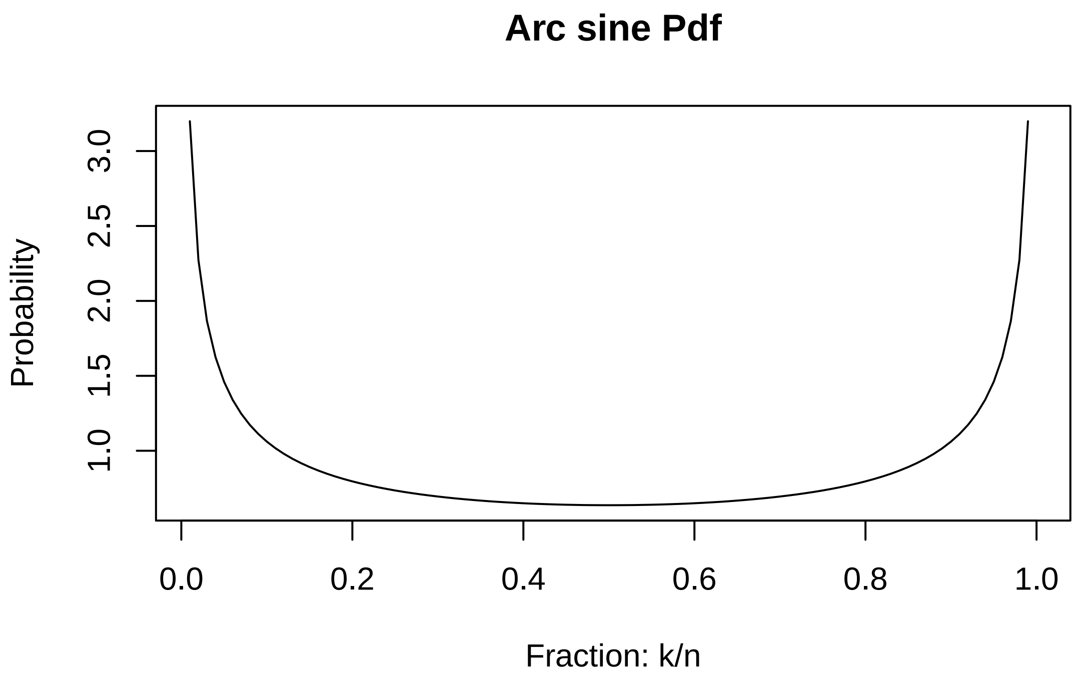

Figure 8: Limiting probability density function of a draw for different fractions k/n

To compute the probability that fraction k/n spends in the positive area for a period which is higher than $1/2$ and lower than some fraction α, i.e. $1/2 < \alpha < 1$. According to our derivation this probability is given by

$$\sum_{n/2<k<\alpha n} p_{2k,2n} = \frac{1}{\pi n} \sum_{n/2<k<\alpha n} \frac{1}{\left(\frac{k}{n}(1-\frac{k}{n})\right)^{1/2}}, \tag{8}$$

which can be approximated by a Riemannian integral

$$\pi^{-1} \int_{1/2}^{\alpha} \frac{dx}{\left(x(1-x)\right)^{1/2}} = 2\pi^{-1} \arcsin \alpha^{1/2} - 1/2. \tag{9}$$

And because of the fact that

$$\pi^{-1} \int_{0}^{1/2} \frac{dx}{\left(x(1-x)\right)^{1/2}} = 1/2 \tag{10}$$

we obtain

$$\pi^{-1} \int_{0}^{\alpha} \frac{dx}{\left(x(1-x)\right)^{1/2}} = 2\pi^{-1} \arcsin \alpha^{1/2}, \tag{11}$$

which is a cumulative distribution function of the arc sine distribution. The shape of this function is presented in Figure 9. The shape of this fuction indicates that the fraction of time spent in the positive area is much more likely to be close to zero or one, than to be close to the expected value of 1/2.

```
fraction <- (1:100)/100
plot(fraction, 2/pi * asin(sqrt(fraction)), type = "l", xlab = "Fraction: k/n",
    ylab = "Cumulative Probability", main = "Arc sine Cdf")
```

The cumulative probability distribution function (cdf) is very useful when one needs to read the probability of a less fortunate player to win. Let us come back to the previous example and imagine the random walk identifies with two players who play a game. If a player A is currently winning the random walk visits the positive area. If she is losing, the negative area is visited. In that sense the long period of winning by a particular player is equivalent to fluctuation of a random walk. From the cdf we can compute the probability that corresponds to the situation that a less fortunate player is going to win for a certain proportion of time. This proportion can be denoted by X, where $(X < 1/2)$. Then

$$P_X = 2\pi^{-1} \arcsin X^{1/2}, \tag{12}$$

corresponds to the probability that the fraction of time that the less fortunate player has been in a winning position was smaller than X. As we work with a symmetic random walk, there is always some winning player, (and each of two players can be less fortunate), so in fact we need to multiply P_X by a factor of 2 to account for this symmetry. For instance if we take a fraction of 10%, then the corresponding cumulative probability is given by $2 \times 2\pi^{-1} \arcsin 0.1^{1/2} = 0.4$. This means that we can state that there is 40% probability that the less fortunate player would win in maximally 10% of cases. What is more shocking corresponds to extreme levels of for instance $X = 0.01$. Then

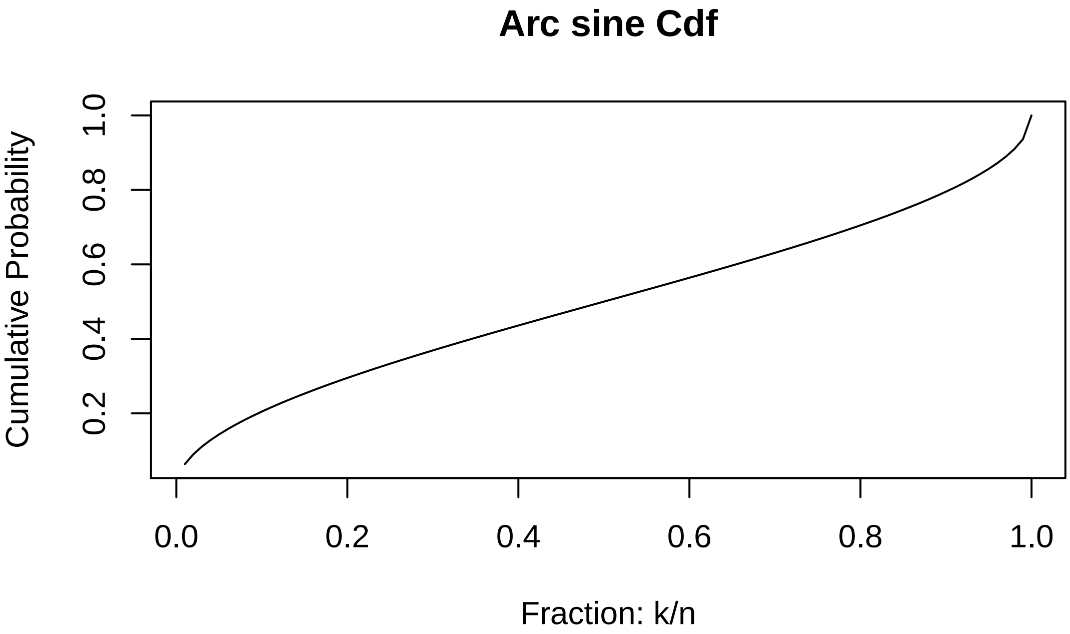

Figure 9: Limiting cumulative probability distribution of a draw for different fractions k/n

$2 \times 2\pi^{-1} \arcsin 0.01^{1/2} = 0.13$. That value implies the 13% probability that the less fortunate player would be winning not more than 1% of cases. This results are striking! Although the pdf and cdf that we work with are based on asymptotic approximations, they work fine even for so low numbers as $n = 10$.

8.2 Simulation study with symmetric random walk

In what follows we present a short simulation study that confirms the result of the theoretical result. We simulate 1000 paths of random walks of length 100. In practice, given one path, for each out of 100 periods it spans we sample either plus one or minus one, both with probability 1/2, and we record sum of such a sequence in every period, what results in random walk path. We repeat the process for 1000 trajectories of the random walk. Then, for each trajectory, we count a number of periods that the random walk has been observed in the positive trajectory. Those counts are recorded in a vector. In Figure 10 the blue line corresponds to the cumulative distribution function of this vector. The theoretical counterparty based on analytical form of the respective arc sine cdf is plotted in red. We observe that the approximation is almost ideal. However, this is obtained for relatively long series of 100 periods. As CromoLab platform has great potential to work with short price cycles, we need to check the applicability of the theory to short time series. In what follows we reduce the length of each random walk path to 10 to confirm, if the approximation with the limiting cdf is still valid. The results is presented in Figure 11. We observe that despite of very limited data input, the arc sine law is definitely still binding.

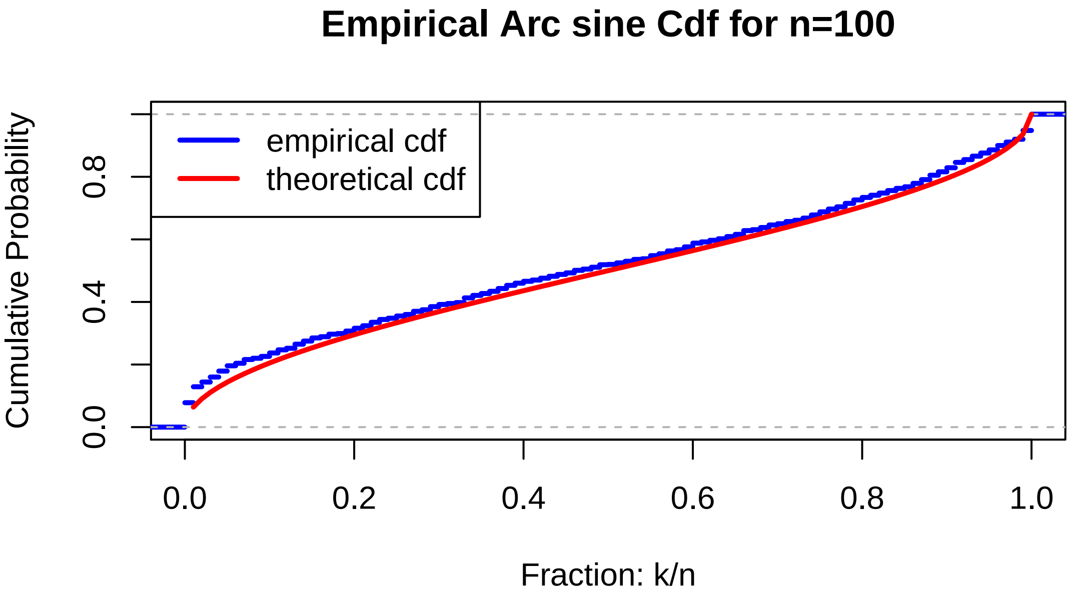

Figure 10: Empirically estimated cumulative probability distribution of a draw for different fractions k/n

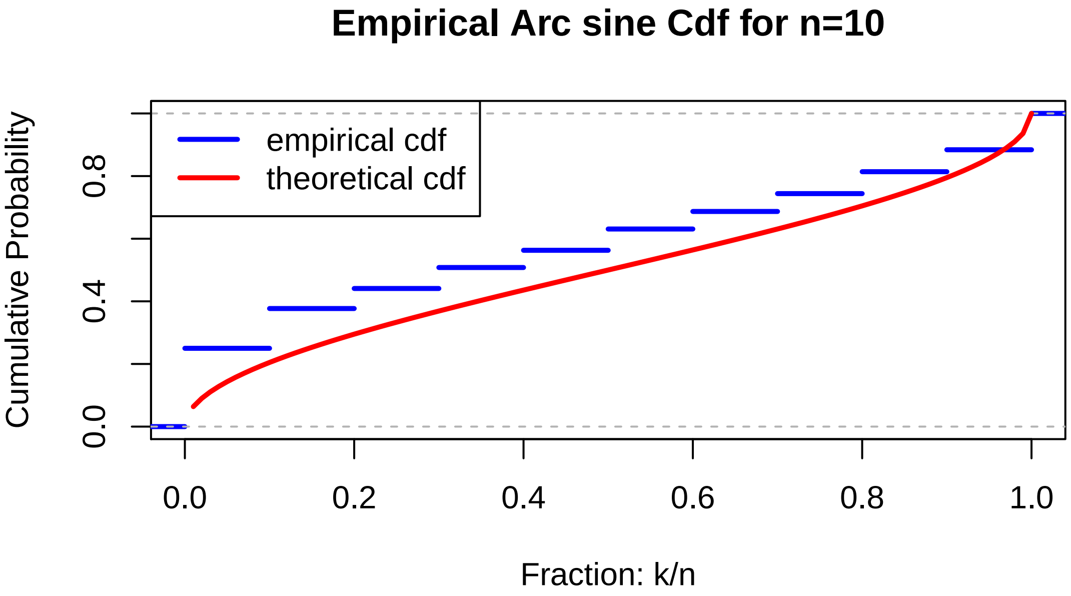

Figure 11: Empirically estimated cumulative probability distribution of a draw for different fractions k/n

8.3 Simulation study with asymmetric random walk

Next we start perturbing the random walk probability distribution. Instead of equal probability of occurrence of plus and minus ones set at 0.5, we consider 10 additional scenarios in each of them increasing the probability of plus one by 0.01. Thus the probabilities of plus one in the alternative scenarios vary from 0.51 to 0.6. For each of those scenarios we repeat exactly the same experiment. We simulate 1000 of random walks of length 100 with a selected level of probability of plus one. Figure 12 presents outcome of this experiment. This is a key plot in this paper and main motivation for CromoLab project. It shows that perturbing the random walk probability by even a few percent, leads to exteme fluctuations in the random walk behavior. The fluctuations are much longer than in case of symmetric random walk with equal probability of up- and downmovement. And even in case of symmetric random walk the fluctuations are substantial. The most bottom curve in Figure 12 corresponds to the experiment dedicated to random walks with probability of up-movement equal 0.6. In this case there is so much probability cumulated around the fractions $k/n > 0.8$, that the probability of random walk to traverse in the negative side is negligible. If we observe a process that is characterized by probability of increase at the levels which are higher than 0.5, betting for such process to be in positive area is almost a sure deal. If we think of cryptocurrency price modelled by random walk, this is exactly the situation. If the probability of increase for the process modeling a given cryptocurrency dominates the probability of decrease, there is a substantial likelihood of exerting abnormal return with a long market exposure in this cryptocurrency. If, on the other hand, the probability of decrease is dominating, we should short sell this cryptocurrency. It seems extremely easy if one knows the probability distribution $(p, 1-p)$ which govern the random walk. But… who knows this probability?

The objective of CromoLab is to develop a model to estimate the probability distribution $(p, 1-p)$ in every moment of time and for each data frequency. In the prototype CromoLab has developed such a model under daily data frequency. In the next development phase we develop this model further at other frequencies. In the following sections we describe the necessary ingredients for development of such a statistical model.

```
# length of random walk
T <- 100
# number of simulations of random walk
nSim <- 1000

randomWalkRecordings <- matrix(NA, nSim, T)
# number of scenarios for probability of plus and minus ones
J <- 11
# beingInPositive is avariable indicating percentage of time that the random
# walk spends in the positive area
beingInPositive <- matrix(NA, nSim, J)
for (j in 1:J) {
    prob <- 0.5 + 1 * (j - 1)/100
    for (i in 1:nSim) {
        eps <- rbinom(T, 1, prob)
        eps[eps == 0] <- -1
        randomWalkRecordings[i, ] <- cumsum(eps)
        beingInPositive[i, j] <- sum(randomWalkRecordings[i, ] > 0)/T
    }
}
plot(ecdf(beingInPositive[, 1]), do.points = F, lty = 1, col = "blue", xlim = c(0,
    1), lwd = 2.5, xlab = "Fraction: k/n", ylab = "Cumulative Probability",
    main = "Empirical Arc sine Cdf for n=10")
for (j in 2:J) {
    lines(ecdf(beingInPositive[, j]), do.points = F, lwd = 1.5)
}
p <- (1:100)/100
cdf <- 2/pi * asin(sqrt(p))
lines(p, cdf, col = "red", type = "l", lwd = 2.5)
legend("topleft", c(`empirical cdf for p/n=q/n` = 1/2, "theoretical cdf", "cdfs based on pertu
    lty = c(1, 1, 1), lwd = c(2.5, 2.5, 1.5), col = c("blue", "red", "black"))
```

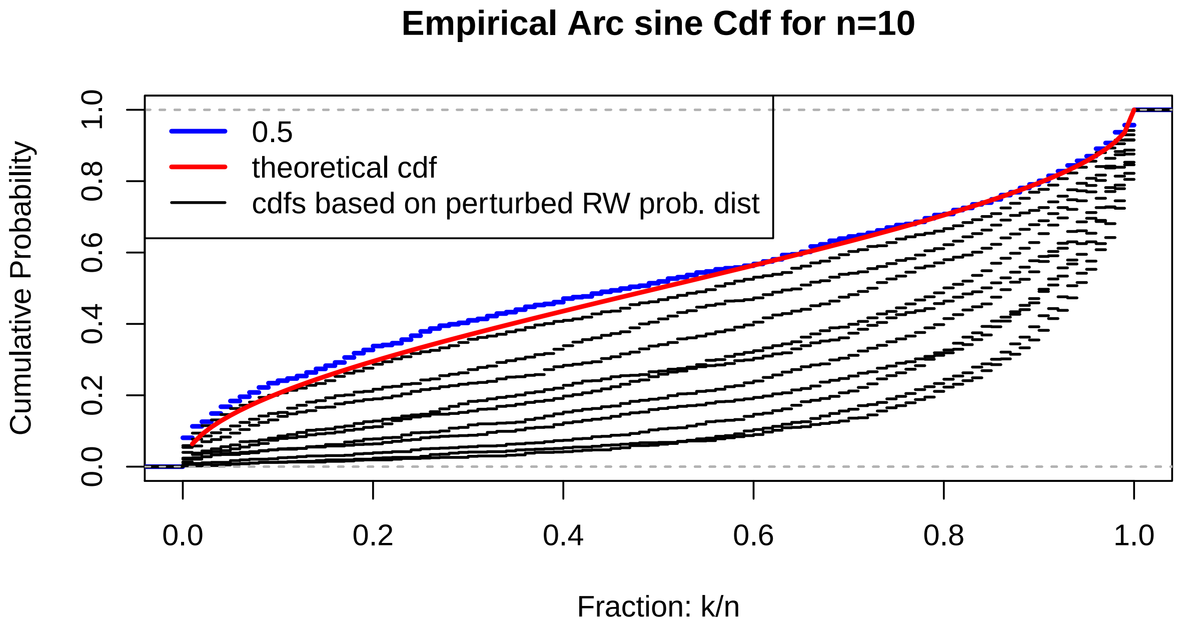

Figure 12: Empirically estimated cumulative probability distribution of a draw for different fractions k/n

9 Cryptocurrency time series model

According to the previous section the cryptocurrency price in CromoLab engine are modeled with random walk. The continuous interpretation of random walk by Brownian motion allows for introduction of some time series features which are important for representation of cryptocurrency price dynamics. Brownian motion should be regarded as a macroscopic picture resulting from a particle moving randomly in d-diemnsional step without making big jumps. In case of cryptocurrency price modeling we work in 1-dimensional space. In spirit of tradition started by for instance (Lévy 1939), we regard Brownian motion as a process with independent increments which are Normal. All the ideas presented above in particular arc sine laws etc. are fully binding in case of Brownian motion, see (Mörters and Peres 2010) Chapter 5 for a detailed exposition.

An important feature which is confirmed in case of cryptocurrency price is heterescedasticity. CromoLab team members have researched this topic extensively, see (Johansen and Gatarek 2017). Heteroscedasticity describes the process characterized by sub-populations (sequences of periods) that have different variabilities from others. By variability we mean the variance of the time series. Although cryptocurrencies can be considered to be relatively new, there has already been some initial analysis into the crypto–currency price data generating process. (Hencic and Gourieroux 2014) applied a non–causal autoregressive model to detect the presence of bubbles in the Bitcoin/USD exchange rate. (Sapuric and Kokkinaki 2014) measures volatility of Bitcoin exchange rate against six major currencies. (Chu, Nadarajah, and Chan 2015) provide a statistical analysis of the log–returns of the exchange rate of Bitcoin versus the USD. They found that the Generalized Hyperbolic distribution seems to be the most appropriate choice to model the unconditional distribution of crypto–currencies time–series. Finally, (Grassi and Catania 2017) find that a robust filter for the volatility of crypto–currencies time–series is strongly required by the data. Moreover, they find evidence of long memory in the volatility for some series, while for others a simpler specification is enough. Finally they find that, differently from foreign exchange currencies, leverage effect has a substantial contribution in the volatility dynamic. On average, volatility increases more after negative shock than after positive shock as in the equity market, hence, crypto–currencies time–series incorporate the leverage effect with some degree of heterogeneity across the series. Finally they find evidence of time–varying skewness for some series and absence of time–varying kurtosis for the whole sample. CromoLab is going to highly explore findings of (Grassi and Catania 2017) in modeling.

10 Time series filtering: estimation of probability evolution

The models applied in case of cryptocurrency modeling need to tackle an important aspect of real-time adaptive modeling, in the sense that inference must be made on-line, before the data collection ends. For those types of applications one must have, at any time, an up-to-date estimate of the current state of the model. By the state of the model, the current value of parameters is considered. In that case we estimate the probability of an price move from the currency price of the assets and as such we need to have the best possible current estimate of the probability. The density of the current estimate of the parameters is usually called the filtering density. Let us denote it by $p(\theta_t|y_{1:t})$ to denote its real time character and conditioning on all the observations in the past, i.e. $1 : t$. This expression denotes the best knowledge about the parameter θ_t that we have based on the information in the data up to the period t. In case of CromoLab statistical engine, the main parameter to be estimated/filtered is the probability of price increase, decrease, so the distribution $(p, 1 - p)$. In this real time context, this expression obtain subscript t to denote its time varying character i.e. we estimate $(p_t, 1 - p_t)$. Although p_t is definitely the most important parameter of the system, there are plenty of other coefficients that the system needs to estimate in the real time. For instance parameters of the cointegration model and hedging parameters. Therefore for expository purposes we work with symbol θ_t to refer to a general character of the approach rather that focusing on probability p_t.

10.1 Bayesian approach to econometrics and its parallel to probability filtering

Time series filtering has a lot to do with discipline of Bayesian econometrics, in particular computational Bayesian econometics. It is a framework which gives basis for flexible estimation of model parameters in particular with real time data streaming. (Van Dijk and Kloek 1978) has brought sequential Monte Carlo techniques, a basis of all modern Bayesian aparatus, to econometrics and made it possible for hardly analytically solvable models to be estimated by computational methods.

Analysis of real data, nonnecessary economic or financial ones, rarely disposes of perfect information on the phenomenon of interest. Even when an accurate deterministic model for the system under study is available, there is always some measurements error, or imperfections. Statistics deals with uncertainty, not only uncertainty about the data, but also uncertainty about the model form. A basic point in Bayesian econometrics is that all the uncertainty related to a phenomenon should be described by means of probability. In this perspective, probability has a subjective interpretation, being a way of formalizing incomplete information that the researcher has about the events of interest. Probability theory prescribes how to assign probabilities coherently. How about the data based inference on probability?

The Bayesian approach postulates learning from experience. The learning process consists of application of probability rules: one simply has to compute the conditional probability of the event of interest, given the experimental information. Bayes' theorem is the basic rule to be applied to this aim. Given two events A and B, the joint probability of $A \ \ \& \ \ B$ to occur is given by $P(A \ \ \& \ \ B) = P(A|B)P(B) = P(B|A)P(A)$, where $P(A|B)$ is the conditional probability of A given B and $P(B)$ is the refferred to as the marginal probability of B. Bayes' theorem is a direct consequence of the above equalites and it says that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{13}$$

The importance of this formula results from its implications for inductive learning process. In the world of Bayesian econometrics, A represents the event of interest for the analyst, B an experimental result which she believes can provide information about A. Given $P(A)$ and having assigned the conditional probabilities $P(B|A)$ of the experimental fact B conditionally on A, the problem of learning about A from the experimental evidence B is solved by computing the conditional probability $P(A|B)$ according to the Bayes' theorem.

The event of interest A in the statistical inference is usually represented by the vector of parameters θ and the experimental result is usually described by the sample of observed data Y. More specifically, based on the knowledge of the problem, the researcher can assign a conditional distribution $p(y|\theta)$ for Y given θ, called the likelihood. $p(\theta)$ expresses the uncertainty on the parameter θ. $p(\theta)$ is usually referred to as a prior distribution. Upon observing $Y = y$, we can apply Bayes' formula to compute the conditional density of θ given y. This density is referred to as posterior distribution

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}, \tag{14}$$

where $p(y)$ is called marginal distribution of Y,

$$p(y) = \int p(y|\theta)p(\theta)d\theta. \tag{15}$$

The marginal dstribution is only a normalizing factor for $p(y|\theta)p(\theta)$, thus we can use the proportionality sign instead of equality relation

$$p(\theta|y) \propto \frac{p(y|\theta)p(\theta)}{p(y)}. \tag{16}$$

Equation 16 is the key equation of Bayesian inference and filtering at the same time.

It says that the posterior distribution is proportional to the product of likelihood and prior. It presents the underpinnings of the Bayesian paradigm which says that the posterior distribution is based upon the prior distribution and the information coming from the observed dataset (likelihood).

10.2 Example of Bayesian inference with conjugate prior

This concept of Bayesian inference is envisaged with an example of beta distribution. The Beta distribution can be understood as representing a distribution of probabilities that is, it represents all the possible values of a probability when we don't know what that probability is together with the likelihood for each of the potential realization of those unknown probability.

The beta distribution is a convenient family of distributions for modeling a proprtion. It can be a different type of proportion. Proportion of women in the population, proportion of assets to fall in terms of value in the stock exchange, proportion of customers who has been directed to a company website from search engine… Examples can be multiplied. In general the more and more uncertainty in the economic systems, the more business analytics managers know that the proportion should be interpreted in terms of its potential span, taking into account the underlying stochasticity, and not in terms of one given value. Proportion of cryptocurrency price increases over specific period of price measurement can varies as well. Thats why we need the beta distribution.

Apart from some normalizing constants, the beta density for the proportion parameter p, probability again, is proportional to

$$b(p) \propto p^{a-1}(1-p)^{b-1},\ 0 < p < 1,$$

where the hyperparameters of a and b are assumed given and they define the shape of the density. The R-code below plots beta density upon different parametrization schemes. It shows how rich the collection of shapes can be given different hyperparameters. Figure 13 presents the variety of beta distributions parameterized properly.

```r
# short routine for parametrized beta density plotting
betaDensPlot <- function(a, b, curveCol) {
    curve(dbeta(x, a, b), from = 0, to = 1, xlab = "p", ylab = "Density", lty = 1,
        lwd = 4, add = TRUE, col = curveCol)
}
# selection of parameters for beta density plots
parametersMatrix <- matrix(c(3, 7, 2, 7, 2, 1.8, 2, 1, 2, 0.8), 2, 5)
# plot the curves

curve(dbeta(x, 3, 9), from = 0, to = 1, xlab = "p", ylab = "Density", lty = 1,
    lwd = 4, main = "Beta density under different parametrization")
curveColors <- c("red", "blue", "brown", "green", "orange")
```

```r
invisible(apply(matrix(1:dim(parametersMatrix)[2]), 1, function(i) {
    betaDensPlot(parametersMatrix[1, i], parametersMatrix[2, i], curveColors[i])
}))
legend("topright", c("a=3, b=9", "a=3, b=7", "a=2, b=7", "a=2, b=1.8", "a=2, b=1",
    "a=2, b=0.8"), lwd = rep(4, 6), lty = rep(1, 6), col = c("black", curveColors))
```

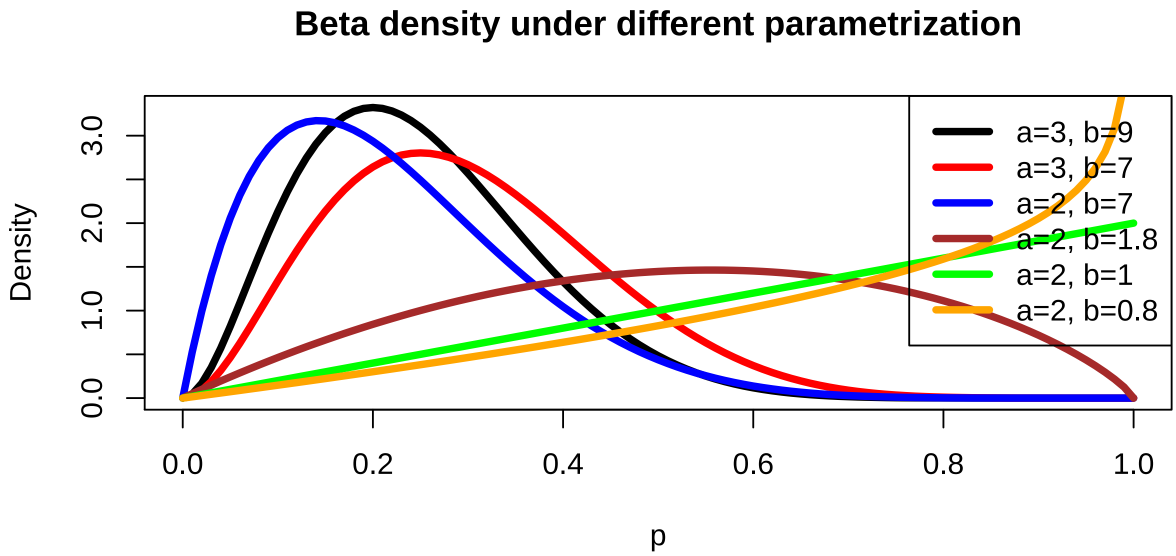

Figure 13: Examples of beta distributions

Let's now get some practical Bayesian statistics done with the beta distribution. The simple example that we show explains the basis of computations to be performed by the CromoLab engine massively. Let's assume we need to solve the problem in which we need to estimate the proportion of an event which we call success. For instance let's assume that a success is associated with a student studying more than 5 hours a day in a library. In case of CromoLab those success will be connected with consecutive number of increases of the price of particular cryptocurrency. To make it easier for now let's adhere to the lazy students example. For the start we assume that learning activity of 30 students is measured from which 12 study more than 5 hours and 18 less than 5 hours. What would be the likelihood function for such a situation? Well, it is a typical situation with a discrete variable x which can be equal to 1 with probability p or to 0 with probability $1-p$, depending if the students works more or less than 5 hours. It does not differ to much from a price growing with probability p and declining with probability $1-p$, only the interpretation is different. The likelihood function for a student is given by

$$l(x) = p^x(1-p)^{1-x}.$$

Thus, if a student works longer than 5 hours, then $x = 1$, and if shorter, then $x = 0$. The likelihood for a single observation can basically be equal to p or $1-p$, which is very logic as the probability of a randomly selected student to work more than 5 hours is equal to p and to belong to the group of the lazy ones to $1-p$.

What is the likelihood for n students? Apparently, due to the independence in the behaviour of the students, the likelihood for the entire population can simply be defined as a product of the likelihoods for the individual students. Thus we can state

$$L(X) = \prod_{i=1}^{n} p_i^x (1-p)^{1-x_i},$$

with $X = (x_1, x_2, \ldots, x_n)$. In case of our example with 12 ambitious and 18 lazy students, this likelihood results in

$$L(p) = p^{12}(1-p)^{18}.$$

Please take into account that the function can now be defined as a function of p. In fact, please note that this likelihood is nothing else than the beta density that we have just introduced above, with $a = s + 1 = 13$ and $b = f + 1 = 19$. To keep the notation in order, we can replace the equality sign with the proportionality one

$$L(p) \propto p^{12}(1-p)^{18}.$$

If we were to consider it in the general terms of an experiment which can result in a success or a failure, and denote by s the number of successes and by f the number of failures, we would obtain the likelihood in the following form

$$L(p) \propto p^{s}(1-p)^{f}.$$

What about combining the likelihood with the prior? In Bayesian statistics, the convenient approach is to work with the prior distribution which belong to the same family of distributions as the likelihood function. That is often referred to as conjugacy. In case of beta distribution the conjugate prior for our beta likelihood is beta distribution as well. Let's assume the prior

$$b(p) \propto p^{a-1}(1-p)^{b-1}.$$

Then, if we go back to the definition of posterior as a product of prior and likelihood, we obtain

$$b(p|data) \propto p^{a-1}(1-p)^{b-1}p^{s}(1-p)^{f} = p^{a+s-1}(1-p)^{b+f-1},$$

which again defines a beta distribution. In this case it is parametrized with $a+s$ and $b+f$. We have just experienced a form of conjugacy in its purest form. The prior and the posterior distributions stem from the same distribution family and differ only with respect to the parameter values.

Let us visualize this situation in form of the prior updating with the likelihood information. We assume that the shape of the prior density is determined by $a = 3$ and $b = 7$. This prior distribution, which expresses uncertainty of the researcher around the true value of probability parameter p (x-axis), is enriched by the information from the dataset (also in form of beta density) to result in a posterior distribution in the same family of distributions. The R-code presenting prior update is presented below.

```r
# parameters of prior distribution
a <- 3
b <- 7
# parameters of the likelihood
s <- 12
f <- 18
curve(dbeta(x, a + s, b + f), from = 0, to = 1, xlab = "p", ylab = "Density",
    lty = 1, lwd = 4, main = "Prior updating and beta conjugacy")
curve(dbeta(x, s + 1, f + 1), from = 0, to = 1, xlab = "p", ylab = "Density",
    lty = 2, lwd = 4, add = TRUE)
curve(dbeta(x, a, b), from = 0, to = 1, xlab = "p", ylab = "Density", lty = 3,
    lwd = 4, add = TRUE)
legend("topright", c("Prior", "Likelihood", "Posterior"), lty = c(3, 2, 1),
    lwd = c(3, 3, 3))
```

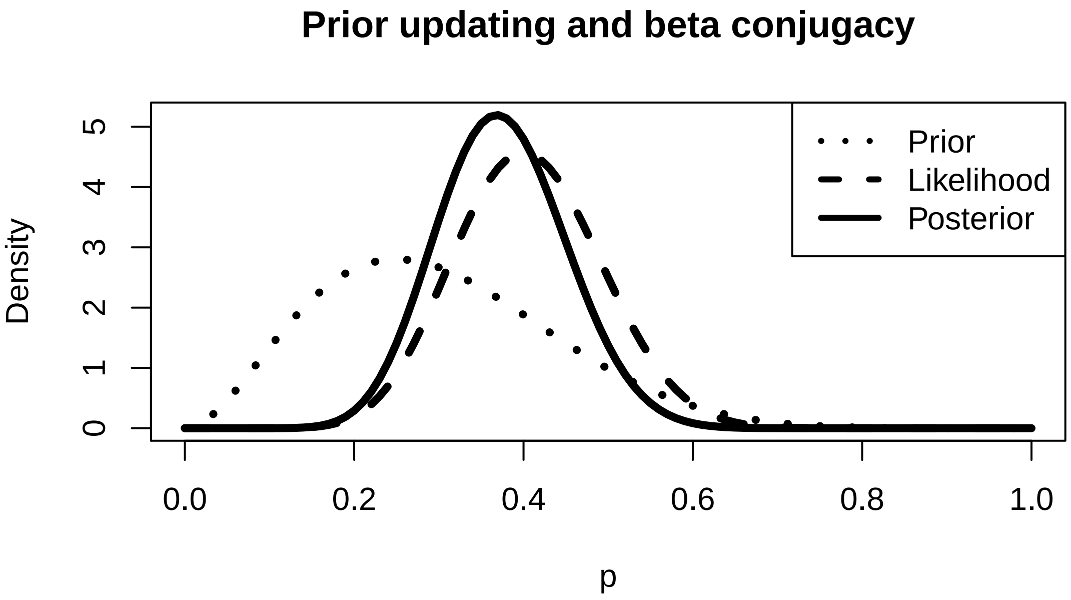

Figure 14: Bayesian updating

The shape of posterior in Figure 14 is much closer to the likelihood than to the prior, which seems natural and in line with the parameters. We have just seen, how easy it is to infer on the proportion if we deal with the information about it combined from two sources: some external source, which impacts our inference - prior, and internally collected data - likelihood. This example is very representative of methodology that we are going to apply in CromoLab platform statistical engine. The inductive learning process which constitutes the basis of our inference is a main building element of the philosophy of CromoLab. We try to apply simple statistical model form but to explore the data to the maximum with smart computational inference techiques. Thats the objective of CromoLab.

10.3 Example of Bayesian inference with discrete prior

To make a step forward. We assume that our prior is not given in terms of a closed form distribution, but it is just defined in terms of some believes regarding probability of possible values. Is this important? It is actually extremely important for CromoLab. CromoLab bets for massive analysis of blockchain transacitonal data. Patterns which are observed in the system shape the beliefs about probability distribution. Analysis based on the past data will be informative for inference with new stream of information. The more mature the system gets in terms of abundance of information that it had processed, the less it is going to adhere to the analytical distributional forms for the parameter inference.

To explain this concept with an example, let us continue with the previous example. We shall assume that we have a vector of believes that proportions

$$0.1, 0.2, 0.4, 0.5, 0.7$$

are possible values for probability p. These values are assigned corresponding weights. For instance based on the frequency of observations of the proportions

$$1, 4, 5, 2, 5.$$

We can convert this vector into the vector of prior probabilities just by dividing each weight by the sum of them. Let us see that in action in Figure 15.

```r
# discrete prior: believes on proportions
p = c(0.1, 0.2, 0.4, 0.5, 0.7)
# discrete prior: frequency of proportions
priorProb <- c(1, 4, 5, 2, 5)
priorProb <- priorProb/sum(priorProb)
# prior
plot(p, priorProb, type = "h", lwd = 5, xlab = "p", ylab = "Prior probability",
    main = "Discrete prior")
```

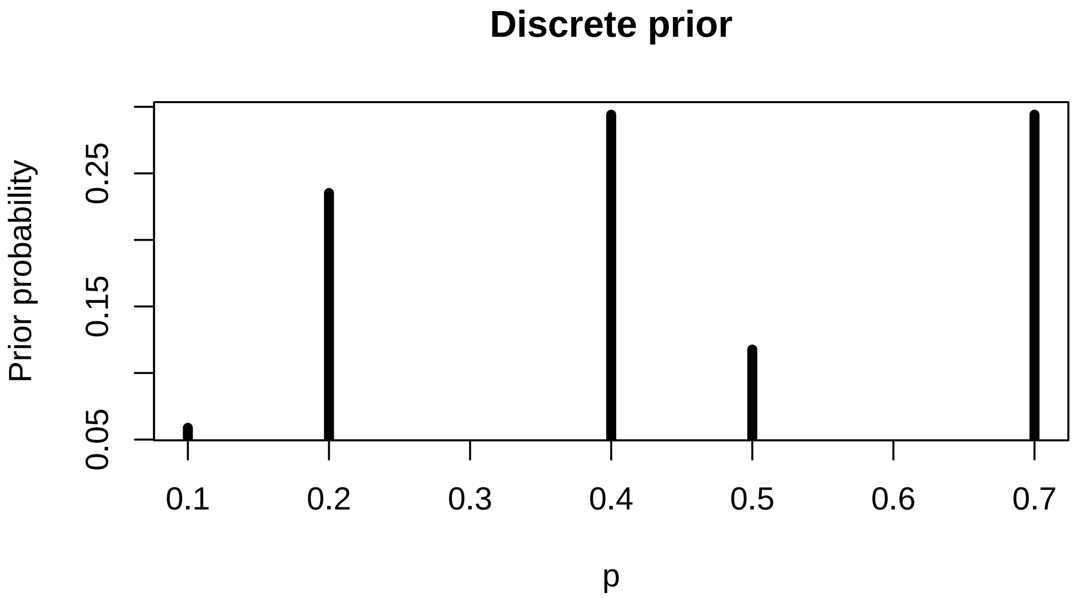

Figure 15: Example of discrete prior

Figure 15 shows histogram of this distribution. This is a typical discrete distribution. How to input this knowledge into the analysis? Well, it is fairly simple. We just need to multiply those values with the corresponding values of likelihood function evaluated for the vector of believes with regards to proportion. The implementation is presented in Figure 16.

```r
# discrete prior: believes on proportions
p = c(0.1, 0.2, 0.4, 0.5, 0.7)
# discrete prior: frequency of proportions
priorProb <- c(1, 4, 5, 2, 5)
priorProb <- priorProb/sum(priorProb)

# parameters of the likelihood in beta form
s <- 12
f <- 18

# evaluation of parameters for the believes
like <- s * log(p) + f * log(1 - p)
# compute the evaluations of posterior
product <- exp(like) * priorProb
# make a proper denisty out of the posterior
postProb = product/sum(product)

library(lattice)  # for plotting
# data frame for prior
prior <- data.frame("prior", p, priorProb)
# data frame for posterior
posterior <- data.frame("posterior", p, postProb)
names(prior) <- c("type", "p", "probability")
names(posterior) <- c("type", "p", "probability")

# combining the information into one data frame
data <- rbind(prior, posterior)
# plot
xyplot(probability ~ p | type, data = data, layout = c(1, 2), type = "h", lwd = 3,
    col = "black", main = "Discrete prior updating with likelihood")
```

The posterior based on discrete prior is discrete itself. Despite of the continuous likelihood form, the discrete character of the prior leads to the discrete posterior. That is extremely useful for CromoLab inference. Such an estimated posterior can now be used as a prior in the next sequence of the analysis, combining information from another subject. For instance, let us imagine that we have estimated the posterior of proportion of ambitious students at one univeristy. Now, we would like to repeat the same experiment at another one. The posterior, in the discrete form, that we have obtained at the first university can be used as a prior for the analysis at the second one. This way we combine more and more information form different sources. The discrete from of the posterior make it directly transferrable into a discrete prior. That is the way, the data obtained at a previous period of measurement of cryptocurrency price would be applied as a prior for a parameter estimation in an upcoming period, be it a second, hour or a day. This sequence can

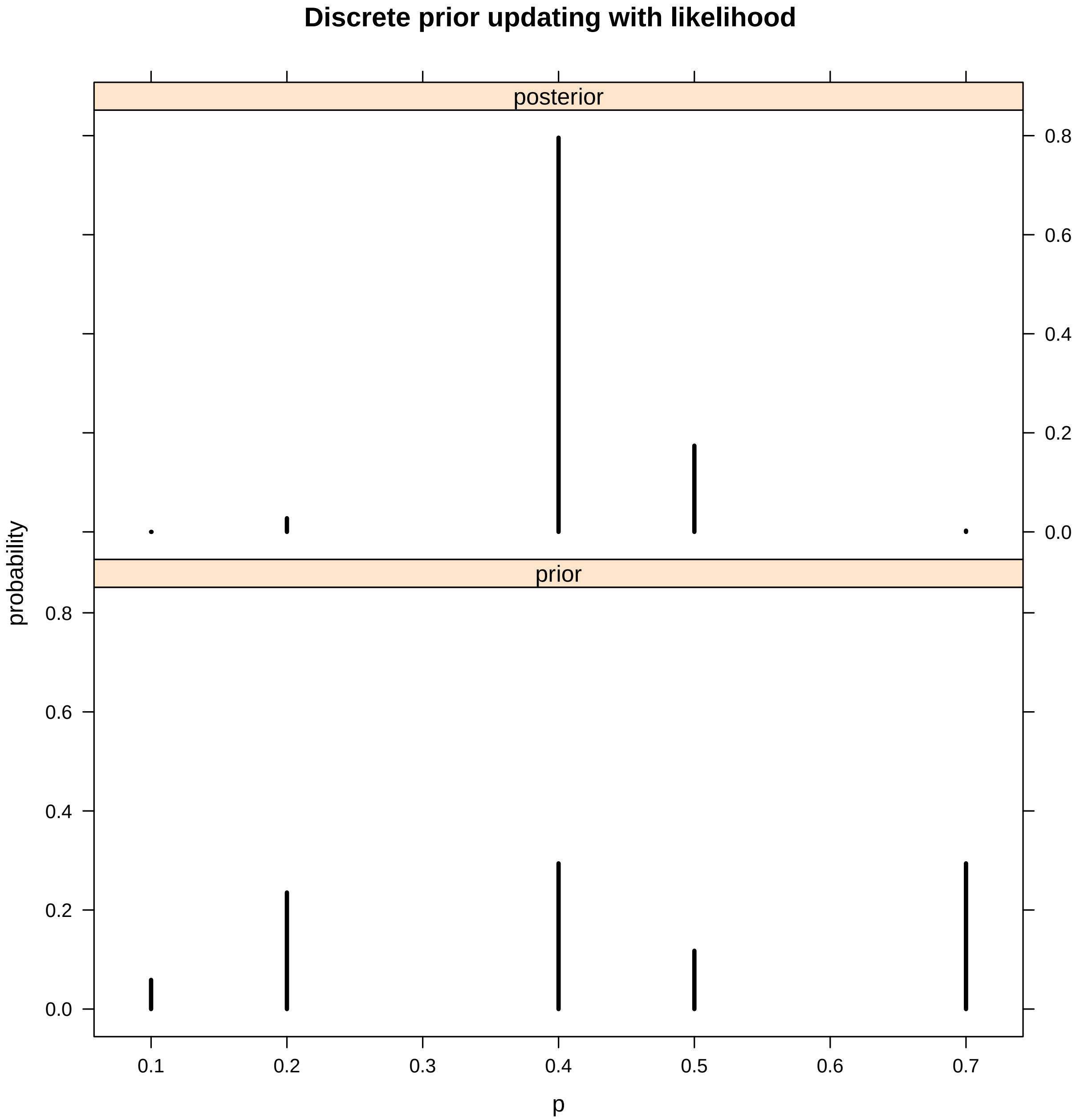

Figure 16: Bayesian inference with discrete prior

continue infinitely. Equation 17 presents diagrammatically the manner in which posterior obtained with the information up to period t becomes a prior for period $t + 1$ etc. All have discrete character.

$$\begin{aligned} Prior_t \times likelihood_t -> posterior_t \\ Prior_{t+1} < -posterior_t \\ Prior_{t+1} \times likelihood_{t+1} -> posterior_{t+1} \\ Prior_{t+2} < -posterior_{t+1} \\ \dots \end{aligned} \tag{17}$$

10.4 Filtering as a process of continuous Bayesian updating

After presenting the mechanics of Bayesian inference, we come back to time series filtering which shall be interpreted as a continuous process of parameters updating according to the Bayesian inductive principle. We should regard it as repeating the examples above continuously with new information coming and the state of the parameters in the previous period being regarded as a prior for new estimate of the parameter state.

Figure 14 and Figure 16 have presented the concept of prior updating with likelihood based information. We see how the density moves from prior to posterior based on the likelihood data in between. The form of posterior is identical with the form of prior irrespective of how the likelihood function is defined. If prior is given by a continuous distribution, the posterior is also continuous. The discrete prior results in discrete posterior. The examples concern a fixed data sample i.e. a sample which is finite and no new observations are streamed. In case of real time analysis of time series, filtering methods, updating of prior repeats every period. The estimates of the parameter in the previous period becomes a prior distribution for the estimate of the parameter in the current period. With the data observed in the current period, the estimate is update what shifts the density of the event of interest accordingly to what has been recorded with the new observation. We discuss it shortly based on formulas of filtering density mentioned above. It is important to understand that abundant information would make the discrete prior almost continuous. Therefore CromoLab bets for data driven inference with as little as neccessary assumptions regarding functional forms.

The density of $p(\theta_t|y_{1:t})$ can derived based on definition of the conditional probability, again,

$$p(\theta_t|y_t, y_{1:t-1}) = \frac{p(\theta_t, y_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \tag{18}$$

At the same time by reapplicability of this formula for conditional probability we obtain

$$p(\theta_t, y_t|y_{1:t-1}) = p(y_t|\theta_t, y_{1:t-1})p(\theta_t|y_{1:t-1}) = p(y_t|\theta_t)p(\theta_t|y_{1:t-1}), \tag{19}$$

in which we recognize the likelihood $p(y_t|\theta_t)$, associated with currently recorded observation y_t, and the prior density $p(\theta_t|y_{1:t-1})$ which describes the prior knowledge about the state of the parameter θ_t before the observation y_t is recorded.

By joining the equations together we can obtain exact form of filtering density

$$p(\theta_t|y_t, y_{1:t-1}) = \frac{p(y_t|\theta_t)p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \propto p(y_t|\theta_t)p(\theta_t|y_{1:t-1}). \tag{20}$$

The proportionality sign $\propto$ refers to no influence of $p(y_t|y_{1:t-1})$ on the shape of $p(\theta_t|y_t, y_{1:t-1})$. It is just a scaling factor which impacts the height of the $p(\theta_t|y_t, y_{1:t-1})$ density. The density $p(\theta_t|y_{1:t-1})$ in Equation 20 should be described as a one-step-ahead predictive density for the states. Given where the system was in the previous period represented by the density $p(\theta_{t-1}|y_{1:t-1})$, and given that we know the form of dependence of state of the parameter in the current period as a function of the state of the parameter in the previous period $p(\theta_t|\theta_{t-1})$, we can derive

$$\begin{aligned} p(\theta_t|y_{1:t-1}) = \int p(\theta_t, \theta_{t-1}|y_{1:t-1})d\theta_{t-1} = \int p(\theta_t|\theta_{t-1}, y_{1:t-1})p(\theta_{t-1}|y_{1:t-1})d\theta_{t-1} \\ = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|y_{1:t-1})d\theta_{t-1}. \end{aligned} \tag{21}$$

Thus $p(\theta_t|y_{1:t-1})$, which is interpreted as a prior gives some vision on where potentially the system can go in the current period, based on where it was in the previous epoch. This vision refers to the situation before the new observation enters into the environemnt. The measurement of data bringing this new observation revises this vision and corrects it accordingly to where the data direct the parameters of the system.

The techniques of time series filtering constitute basis for filtering of probability distribution of random walk which models the price of the cryptocurrencies. What remains to be explained is the methods of operationalizing this idea in empirical application. In case of fixed data sample examples that we have presented in the previous sections, the posterior resulted from multiplication of prior and posterior. In this case the process is sequential. Posterior from previous period shall be regarded as a prior for the incoming period. To implement this idea We apply the method of particle filtering. The project assumes applying this technique at all possible level of data granularity, starting from daily prices and going deeper and deeper to obtain full understanding of the probability distribution behind the ups and downs of the system under study.

10.5 Particle filtering

In many simple applications with Normally distributed variables the usual way of filtering is by means of Kalman filter, which is provided based on closed form analytical expressions for the filtering distribution. We refer the reader to (Durbin and Koopman 2012) for introduction to Kalman filtering techniques. In case of the technology developed by CromoLab this approach is not satisfactory due to highly nonnormal data and, most of all, data which have discrete character and needs to be modeled by means of discrete distributions (Binomial, Poisson and Negative-Binomial). (Durbin and Koopman 2012) presents methods of linearisation of nonlinear and Non-Gaussian processes and methods of importance sampling for such processes. The simultions techniques they consider have been previously considered for likelihood parameter estimation in (Shephard and Pitt 1997). Methods presented in (Shephard and Pitt 1997) has resulted in invention of particle filters in econometrics in (Shephard and Pitt 1999). This method has got applicable for many analytically unsolvable models. CromoLab applies many discrete models in its statistical engine. Therefore this methodology is so broadly applied by CromoLab. For instance the probability distribution of random walk modeling the data is filtered under assumption of Negative-Binomial distribution for the process of forming up- and downmovement in the cryptocurrency price process. The probability underlying this process is time varying. CromoLab filters the probability underlying this distribution with particle filtering approach. Particle filtering is how sequential Monte Carlo is usually referred to in applications to state space model. It is easiest to understand when viewed as an extension of importance sampling, what was a reason why insights in (Shephard and Pitt 1997) has led to (Shephard and Pitt 1999). For this reason this subsection is started with a short discussion of importance sampling. The main difficulty of filtering is to evaluate the integrals entering the formulas in the filter as for instance in Equation 20.

To present the idea of importance sampling, let us suppose that we are interested in evaluating the expected value of some function $f(X)$

$$E_\pi(f(X)) = \int f(x)\pi(x)dx. \tag{22}$$

Due to difficulty of sampling the target density π, we sample another density, g, known as an importance density having the property that $g(x) = 0$ implies $\pi(x) = 0$, then

$$E_\pi(f(X)) = \int f(x)\frac{\pi(x)}{g(x)}g(x)dx = E_g(f(X)w^*(X)), \tag{23}$$

where $w^*(x) = \pi(x)/g(x)$ is the so-called importance function. This suggests approximating the expected value of interest by generating a random sample x^i, $i \in 1, \ldots, N$ from g and computing

$$\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})w^*(x^{(i)}) = E_\pi(f(X)).$$

The sample x^i, $i \in 1, \ldots, N$, with the associated weights w^i, $i \in 1, \ldots, N$, can be viewed as a sample from target density π. This is a great idea which has been applied in Bayesian econometrics over more than 3 decades after (Van Dijk and Kloek 1978) has brought importance sampling to econometrics. Then the modern, computational, econometrics has begun.

In filtering problem, the main challenge concerns the target distribution which changes every time a new observation is made, basically every single period, moving from $p(\theta_{0:t-1}|y_{1:t-1})$ to $p(\theta_{0:t}|y_{1:t})$. How to efficiently update the former to get the proper estimate of the latter? Actually we follow the logic explained in formulas Equation 20 and Equation 21. We see that the prior for new period is necessary to obtain the posterior accordingly to Equation 20, denoted as $\theta_t|y_{1:t-1}$. We follow logic in Equation 21 to update this prior properly. To that end for each θ^i_{t-1} from the support of $\hat{\pi}_{t-1}$ we simulate θ^i_t according to $p(\theta_t|\theta_{t-1})$ in Equation 21. This simulated values are often referred to as predicted states (together they from predicted paths). Then based on the simulated candidate for new set of parameter values, we update the system of weights, w^i_{t-1} to w^i_t. Those weights together with the simulated θ^i_t constitute an proper approximation of $\hat{\pi}_t$. Sampling from this distribution according to the weights results in filtered states (the sequence of filtered states over time constitute filtered path).

What remains to be explained is the way to approximate the weights. Dropping the superscripts for notational simplicity, the weights are given by

$$w_t \propto \frac{\pi(y_t|\theta_t) \cdot \pi(\theta_t|\theta_{t-1})}{g_{t|t-1}(\theta_t|\theta_{0:t-1}, y_{1:t})} \cdot w_{t-1}. \tag{24}$$

We refer to (Shephard and Pitt 1999) for the details. These weights need to be normalized to lead to a proper density $\hat{\pi}_t$.

Below we present a example of implementation of this algorithm for binomial distribution. It is assumed that the each period a random variable is drawn from the binomial probability. There is another random variable behind which is modeled as random walk with i.i.d. error terms. This variable is associated with parameter of the binomial distribution by means of some simple transformation. In the simulation study we test particle filtering in this setup. We simulate time series according to the time varying binomial model specification and then we apply particle filter to retrieve the true process from the simulated data. Figure 17 presents the data generating process and its filtered counterpart. On the top of that we present all the predicted and filtered paths. The filtered path in top panel is an average from the path presented in the middle panel.

```r
## define the data generating process define vector of probabilities driving
## the binomial over time
T <- 100  # length of time series
sigma <- 0.2  # std. dev of a variable representing a latent process driving the probability
theta <- cumsum(rnorm(T, 0, sigma))  # the latent process behind the probability
probabilities <- 1/(1 + exp(-theta))  # from latent porcess to probability
## simulate the time varying binomial
Q <- 100  # set the parameter of the binomial
# simulate the data
Y <- matrix(NA, T, 1)
for (i in 1:T) {
    Y[i] <- rbinom(1, Q, probabilities[i])
}

## run the particle filter
nSim <- 10000  # length of chain
thetaPrior <- 0  # prior parameter

# allocate memory to store the information on the simulated probability
# paths
storedPredictedThetaParticles <- matrix(0, T, nSim)
storedFilteredThetaParticles <- matrix(0, T, nSim)

# in the period t=1, the prob0 is necessary and it is assigned from the
# prior of probability, which is supposed to be known (for instance gamma)
storedFilteredThetaParticles[1, ] <- thetaPrior

# allocate memory for particle filter weights
weights <- matrix(0, T, nSim)
storedWeights <- matrix(0, T, nSim)
# the weights for the first period are equal to 1/nSim
weights[1, ] <- 1/nSim

# predicted paths (from theta_t|theta_t-1 distribution)
storedYPredPaths <- matrix(NA, T, nSim)
# number of particles surviving after weights get updated
numberOfUniqueParticles <- matrix(NA, T, 1)
```

```
drawFromCandidate <- rnorm(nSim * T, 0, 0.4)
drawFromCandidate <- matrix(drawFromCandidate, T, nSim)
for (t in 2:T) {
    # loop over time loop over the particles
    for (i in 1:nSim) {
        storedPredictedThetaParticles[t, i] <- storedFilteredThetaParticles[t -
            1, i] + drawFromCandidate[t, i]
        storedFilteredThetaParticles[t, i] <- storedFilteredThetaParticles[t -
            1, i] + drawFromCandidate[t, i]  # candidate density drawing; `

        # evaluate the likelihood
        likelihood <- Y[t] * storedPredictedThetaParticles[t, i] - Q * log(1 +
            exp(storedPredictedThetaParticles[t, i])) + log(choose(Q, Y[t]))
        # the proposal density is different from the state density
        weights[t, i] <- weights[t - 1, i] * exp(likelihood) * dnorm(drawFromCandidate[t,
            i], 0, 0.2)/dnorm(drawFromCandidate[t, i], 0, 0.4)
    }
    weights[t, ] <- weights[t, ]/sum(weights[t, ])
    # sample indexes of the trajectories which are supposed to survive given
    # their weights
    toSurviveParticles <- sample(1:nSim, nSim, replace = TRUE, weights[t, ])

    # Here, we replaced only last 4 particles on the path to be fine with the
    # theory of a path dependence but at the same time not leading to
    # degenertion in the past. If this gets replaced by
    # storedBetaParticles(1:t,:) = storedBetaParticles(1:t,toSurviveParticles);
    # the paths in the past of the process get degenerated to one (common) path
    # storedBetaParticles(1:t,:) = storedBetaParticles(1:t,toSurviveParticles);
    storedFilteredThetaParticles[max(1, t - 4 + 1):t, ] <- storedFilteredThetaParticles[max(1,
        t - 4 + 1):t, toSurviveParticles]
    numberOfUniqueParticles[t] <- length(unique(storedFilteredThetaParticles[t,
        ]))

    storedWeights[t, ] <- weights[t, ]
    weights[t, ] <- 1/nSim
}

# to make simple things faster
namean <- function(x) {
    mean(na.omit(x))
}

# save the output
outputOfFiltering <- list(storedWeights, numberOfUniqueParticles, toSurviveParticles,
    storedFilteredThetaParticles, storedPredictedThetaParticles)

par(mfrow = c(3, 1), mar = c(2, 1, 1, 1))
```

```
plot(apply(outputOfFiltering[[4]], 1, function(x) {
    namean((x))
}), col = "red", lwd = 5, type = "l", main = "True and filtered processes")
lines(theta, col = "black", lwd = 5)
legend("topright", c("filtered process", "true process"), lty = 1, lwd = c(5,
    5), col = c("red", "black"))
matplot(outputOfFiltering[[4]][, 1:100], type = "l", main = "Example of filtered paths")
matplot(outputOfFiltering[[5]][, 1:100], type = "l", main = "Example of predicted paths")
```

11 Portfolio construction for cryptocurrency market

Estimation of probability which is associated with random walk that models cryptocurrency price allows for identifying cryptocurrencies with specific behavior: rising or declining trend. As this step has been accomplished, an investor has two options: either to invest in this underlying cryptocurrency in an outright manner (creating a portfolio consisiting only of this cryptocurrency), or investing in a wider portfolio, consisting also of cryptocurrencies, which constitute a hedge for the underlying crptocurrency. Consider the situation that an investor enters an unhedged, outright position. She is worried about the risk due to changing prices which might developed in an unpredictable manner. This unexpected behavior can be related to some idiosyncratic shocks which are definitely observed in the market and add on substantially to the risk of such an outright investment. To mitigate this risk the investor decides to hedge by short selling other cryptocurrencies which are correlated with the underlying cryptocurrency. Correlation is a measure which represents the tendency of two, or more random variables, to move in the same direction. If applied to the cryptocurrency prices, the positive correlation would imply the tendency for the prices to develop in the same direction. Negative correlation indicates that the cryptocurrency prices express the tendency to move in the opposite directions. Correlation is traditionally applied in hedging portfolio construction. Hedging is applied to reduce or eliminate the systematic risk related to an outright position. There are to main objectives of hedging: making return on a portfolio invariant to market factor (idiosyncratic shocks driving the entire market in unexpected direction) and to minimize the variance of the portfolio. (Johansen and Gatarek 2017) has shown that to properly hedge the assets which developes as random walks evolving accrodingly to some common trends, the investor is better off is she hedges with assets which are not only correlated but also cointegrated with the underlying.

Cointegration refers to a specific econometric model which measures a long term relation between the assets. It is applied to random walks who constitute a very general model of randomness, which, as has been discussed in detail in the previous sections. properly represents the variation of cryptocurrency price over time. Cointegration stands in opposition to the correlation which measures the short term properties of the prices. The methodology developed by (Johansen and Gatarek 2017) shows how to effectively balance the correlation and the cointegration to deliver the information on proper amounts of hedging assets in the portfolio to be held for specific period.

The objective of CromoLab platform is to deliver the hedge ratios, that should be applied for selected trending cryptocurrency and associated portfolio of hedges, in order to optimally hedge the risk related to market price variation (as measured by conditional portfolio variance). In case of rising trend, instead of holding a long position in the underlying asset only, we are buying it and short selling the hedging assets. In case of a short position in the underlying asset (i.e. when the

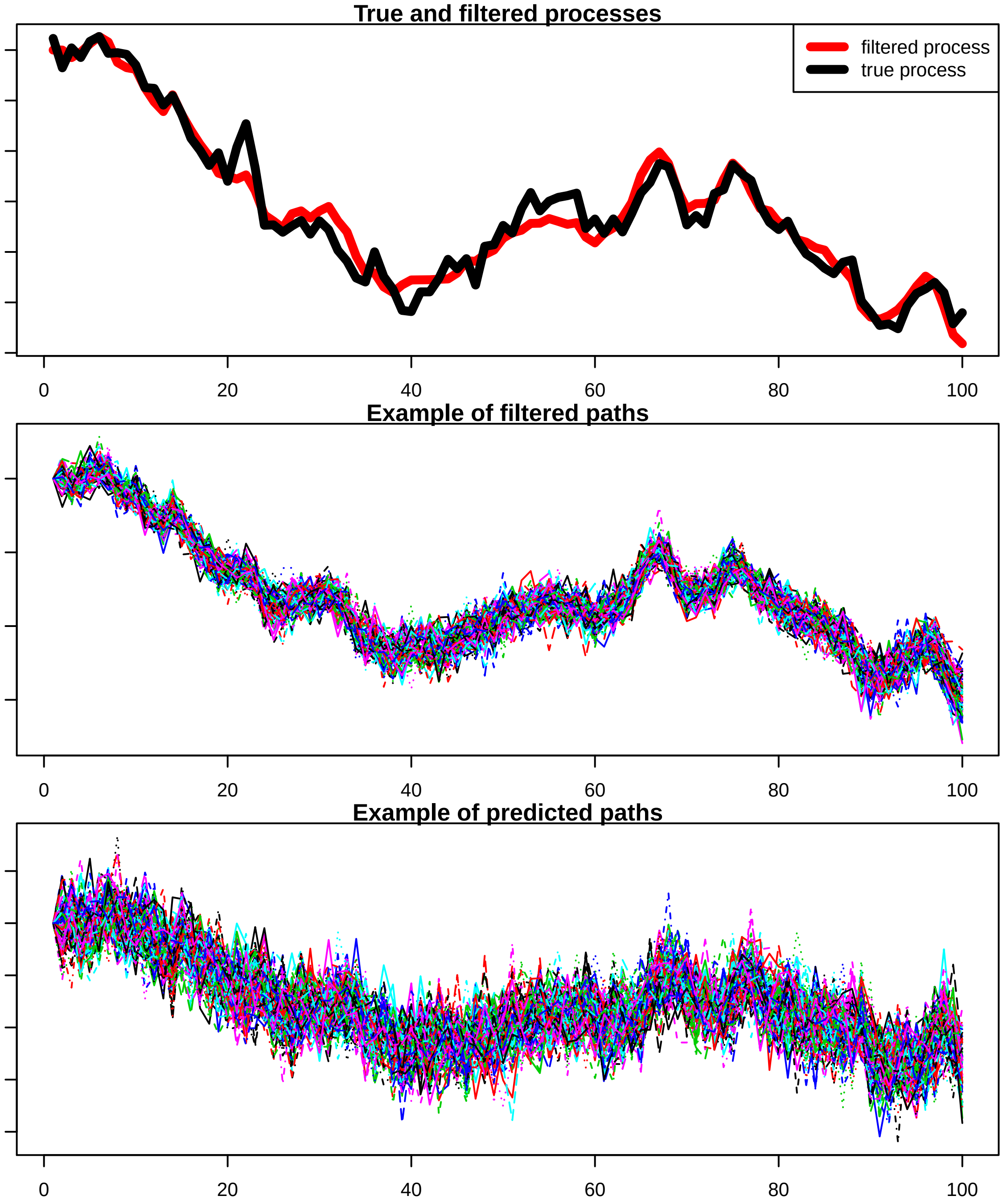

Figure 17: Implementation of particle filtering

trend is declining), we are short selling the underlying asset and buying the hedges according to the hedge ratios.

11.1 Cointegration modeling

Before we present the methodology of (Johansen and Gatarek 2017) which constitute a basis for portfolio construction part of the platform, we shortly explain the idea behind the cointegration model, which is a necessary element on the way to compute the hedge ratios.

The cointegrated vector autoregressive model (CVAR) is assumed to describe the variation of cryptocurrency prices. This model puts forward that random walks, which are nonstationary process, enter in relation between each other to form stationary processes. Thus cointegration models time series which follow common trends. Linear combination of individual time series is nothing more as a common trend which is followed by the individual random walks (representing prices) with a different strentgh. Some of the first attempts to model common trends with cointegration can be found in (Kasa 1992). Since then, cointegration has been found and tested for in many financial markets.

The cointegration approach to trading in financial markets has been implemented in many econometric studies, for instance, (Lin, McCrae, and Gulati 2006), (Vidyamurthy 2004), (Gillespie and Ulph 2001), (Alexander and Dimitriu 2005a), (Alexander and Dimitriu 2005b) and (Gutierrez and Tse 2011). Recently, researcher, who belong to the CromoLab project have co-authored a paper, see (Ardia et al. 2016), which identifies important restriction necessary for application of cointegration to financial market analysis. Those techniques are directly applicable to the case of cryptocurrencies and will be part of the engine behind the platform.

To present an example of how cointegration model works in practice, we have estimated the cointegration model on a set of cryptocurrencies presented in figure Figure 18. This figure presents the evolution of value of Bitcoin and Ripple over last few months, with prices normalized as 1 on the 1st of April 2017. With the red line the common trend estimated with cointegration model has been displayed. It is interesting to observe that the common trend is, usually, smoother than the time series which combine into it.

This estimated example has only expository purposes. In practice of econometric modeling for finance, the cointegration models are large scale and combine many time series in multiple equations. The general outcome of cointegration modeling is a set of parameters which define the number of common trends driving the system of time series. Those parameters are neccessary for implementing hedging methodology presented in (Johansen and Gatarek 2017). This technique is based on mix of correlation and cointegration information. In what follows we present shortly the idea behind this methodology.

11.2 Hedging portfolio for cryptocurrency based on cointegration model

In general, the hedging methods can be divided in two classes: static and dynamic methods. The static hedging techniques assume that the hedging portfolio is selected, given information available in period t, and remains unchanged during the entire holding period $t+1, \dots, t+h$. This is opposed to the dynamic hedging methods which allows for rebalancing the portfolio during the holding period, but we are only concerned with static hedging. The holding period h is implied by the cyclic

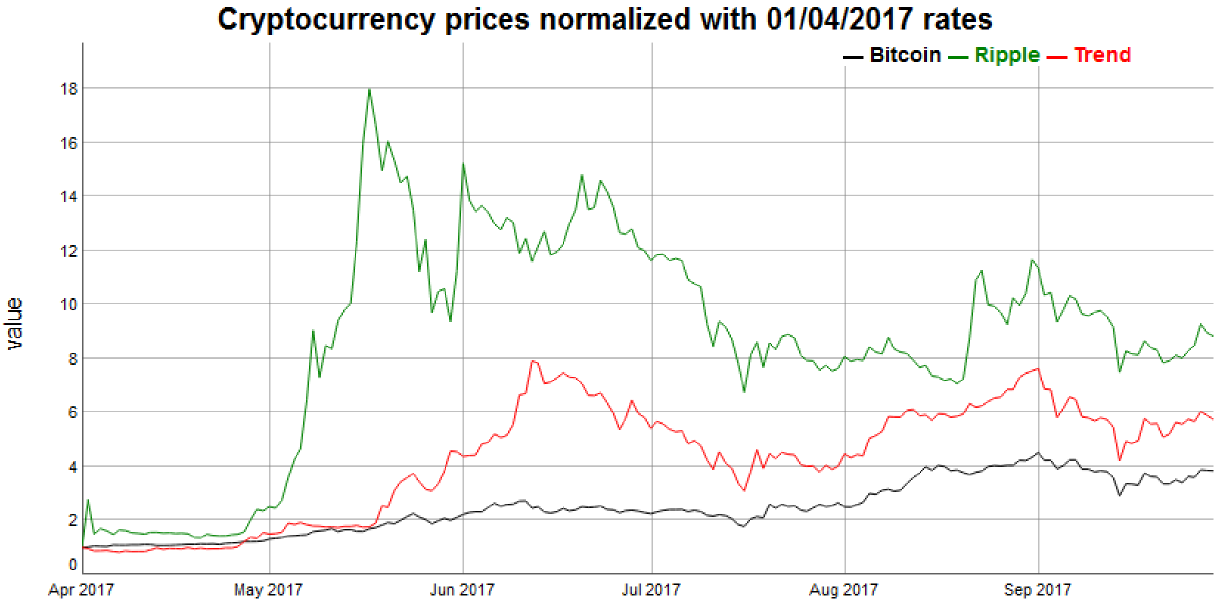

Figure 18: Example of common trend for two selected cryptocurrencies

behaviour in the evolution of probability distribution of random walk. The expected holding period can be derived from properties of this time series.

(Johansen and Gatarek 2017) study optimal hedging for an h-period investment in a system that consists of a set of assets, which follow random walks driven by some common trends. It is assumed that there are n assets, in this case cryptocurrencies, with prices modeled with time series $y_t = (y_{1t}, \dots, y_{nt})'$, and that the first asset is held for h periods, using the other cryptocurrency to hedge the risk, as measured by conditional variance of returns $\Sigma_{t,h} = Var_t(y_{t+h} - y_t)$ given information at time t, that is y_s, $s = 1, \dots, t$.

There are two main results of (Johansen and Gatarek 2017) which find their application in the cryptocurrency analyzing platform, that CromoLab develops. The first set of results concerns derivation of an expression for the risk, $\Sigma_{t,h}$, which depends on conditional (given the period t) volatility of the error term. Based on this expression, the optimal $h-$period hedging portfolio, which minimizes this risk is derived. By the optimal hedging portfolio we mean the weights corresponding to particular cryptocurrencies entering the portfolio. The limit for $h \to \infty$ of the inverse risk matrix, $\Sigma_{t,h}^{-1}$, is found and used to show that the optimal portfolio approaches a variance minimal cointegrating portfolio, which has a bounded risk.

Thus for longer horizons we should choose the variance minimal cointegrating portfolio, which has a bounded risk, and for shorter horizons we should take conditional volatility into account. The period in between constitute a balance between the long term cointegration based hedge and the short term correlation hedge, closely connected to the conditional volatility.

This result is crucial and stands in opposition to the literature in financial econometrics so far, which has positioned correlation as a main source of insight for hedging. According to the statistically based technology developed in (Johansen and Gatarek 2017), the correlation is only informative as soon as one-day ahead holding period is concerned.

The main implications of this methodology is presented in Figure 19. The variance of a portfolio

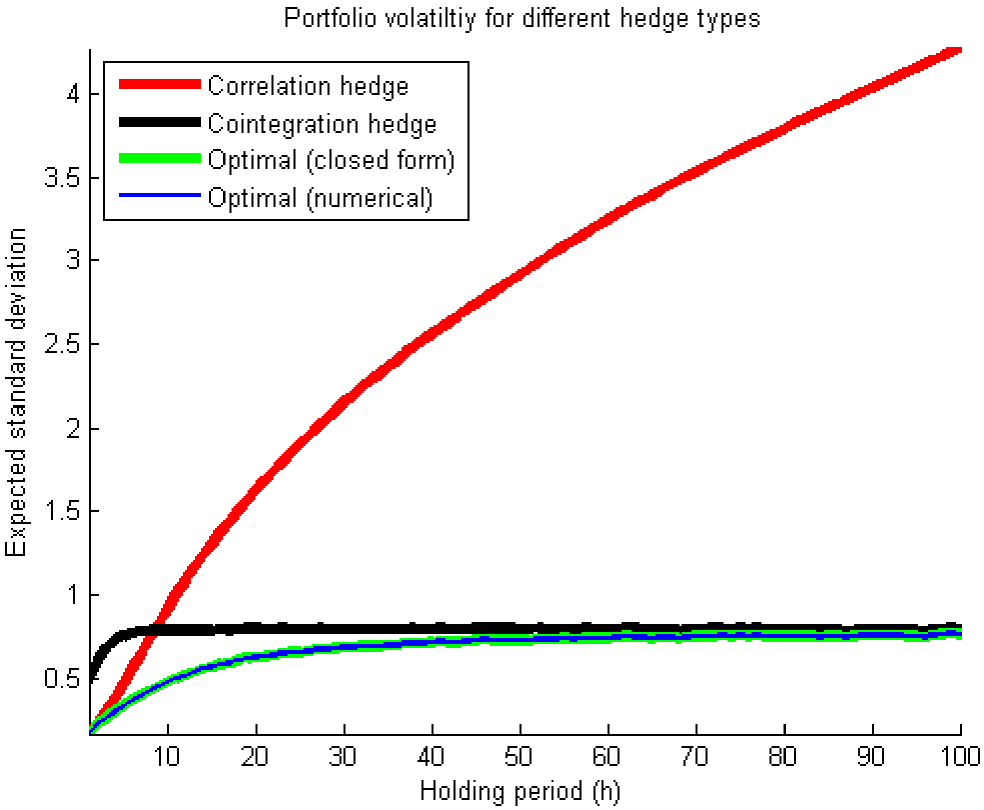

Figure 19: Optimal hedging according to Gatarek and Johansen (2017) versus alternative solutions in terms of minimizing variance (expected standard deviation of a portfolio)

under various hedging methods is presented. The variance of this portfolio is unbounded if the correlation based hedge is applied. At the same time the cointegration based hedge is inefficient for shorter holding period. In that case the correlation hedge is advantageous. Figure 19 is based on a portfolio which consists of two assets only. In case of multiple assets entering the portfolio the extent of inefficiency corresponding to the unoptimal hedging is substantial. First of all it can lead to losses in a consequence of improper hedging (too little hedging assets in the portfolio) as well as inflated transaction costs implied by overhedging. Overhedging leads to an additional risk in terms of an outright position resulting from too high position in a hedging asset which exceeding exposure necessary for hedging.

The second set of results concerns estimation of risk, and the optimal h-period hedging portfolio based on data $y_t, t = 1, \ldots, T$. Under assumptions on the error term that allows for heteroscedasticity, (Johansen and Gatarek 2017) delivers two important results. First they show that a regression of returns $y_{t+h} - y_t$ on information at time t gives a consistent estimator for Σ_h, and a similar result holds if the CVAR is estimated by reduced rank regression. Next it is shown that a regression of y_{1t} on the other prices and a constant gives a consistent estimator of the optimal limiting hedging portfolio.

The conclusion of this is, that if the conditional variance is used as risk measure, in the case of conditional volatility, this has to be modelled by a multivariate GARCH model, like the BEKK model or a multivariate ARCH model. The combined theory of cointegration and a model for heteroscedasticity is challenging. The obvious two-step procedure of first estimating the CVAR assuming i.i.d. Gaussian errors and then use the estimated residuals as input in a BEKK model has not been worked out in details.

The well-known formula

$$Var(y_{t+h} - y_t) = E(Var_t(y_{t+h} - y_t)) + Var(E_t(y_{t+h} - y_t)),$$

shows that the choice between the conditional variance, $\Sigma_{h,t}$ and its expectation, Σ_h, does not involve the variation of the information y_t given at the time of investment.

If a consistent estimator of $\Sigma_{t,h} = Var_t(y_{t+h} - y_t)$ is needed, one has to model conditional volatility, but if the first term $\Sigma_h = E(Var_t(y_{t+h} - y_t))$ can be used, it can be estimated by the simple regression methods or from the CVAR.

The role of cointegration for hedging was analysed by (Juhl, Kawaller, and Koch 2012). They considered a special case of the CVAR and (Johansen and Gatarek 2017) generalizes their results to a CVAR with more lags and more cointegrating relations and allow for a some degree of heteroscedasticity in the martingale error term.

To conclude cointegration plays an important role in hedging. It allows for the possibility that an h-period hedging portfolio has a risk that is bounded in the horizon h, as opposed to the unhedged risk. As important is the result that for moderate horizons, it is important not to use the cointegrating portfolio, but to use the optimal hedging portfolio which interpolates between the short and long-horizon cointegrating portfolio.

12 Portfolio risk analysis

Despite of advanced hedging methods which assure reliable portfolio construction component of the platform the risk can never be fully eliminated. Therefore, the risk analysis component plays important part of the platform. An investor is aware of the risk invoked by the investment. To that end we apply the methodology in (Ardia, Hoogerheide, and Gatarek 2017) that is fully applicable for risk analysis of portfolio with short holding periods. In case of the CromoLab platfrom in development, that constitues an extremely important aspect. Based on initial prototyping and preliminary statistical research we know that the cycles in the cryptocurrency trends extend usually over a few days, maximally to a few weeks. Details of the methodology are presented in the referred paper.

13 Market making with cointegration model

Market making is based on the same cointegration model that is applied to determine the hedging ratios. In that case the cointegration model is used to calculate the statistically correct price of one cryptocurrency based on other cryptocurrencies prices, related with that underlying cryptocurrency. This idea is based upon the assumption that the long term relation holding between cryptocurrencies shall be observed all the time. Any deviation from this relation implies some disequilibrium which is a gap that should be closed soon by market forces (sooner rather than later). Thus, by taking a position in the cryptocurrency whose price deviates from the equilibrium, one is expected to exert a substantial profit. For instance, the investor might wish to open a position in Zcash, which seems to be out of equilibrium, but does not know the fair value of this cryptocurrency. This market as any other on the planet tends to under- and overestimate the value of the cryptocurrencies in some periods of time. They go back to equilibrium quite fast, but if one knew when the price is in disequilibrium, then the profit taking opportunity would be huge. Imagine that according to the cointegration model, the Zcash is in relation to Bitcoin, Ethereum and Dash. If this relation is strong and implied by a stable cointegration model, we can identify the current fair (model implied) price of Zcash based on the current prices of Bitcoin, Ethereum and Dash. If, for some reasons, the Zcash market price deviates from the model implied price, there is possibility for profit. This technique is called market making and it is typically applied by institutional investors, mostly hedge funds, in particular in illiquid assets. CromoLab offers technology for market making in cryptocurrency market.

14 Cromo Lab and IPFS

14.1 What is IPFS

Interplanetary File System (IPFS) is a peer-to-peer distributed file system that connects computing devices with the same system of files. In some ways, IPFS is similar to the Web, but IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content addressed hyper links. This forms a generalized Merkle DAG, a data structure upon which one can build versioned file systems, blockchains, and even a Permanent Web. IPFS combines a distributed hash table, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other, see (Benet 2017).

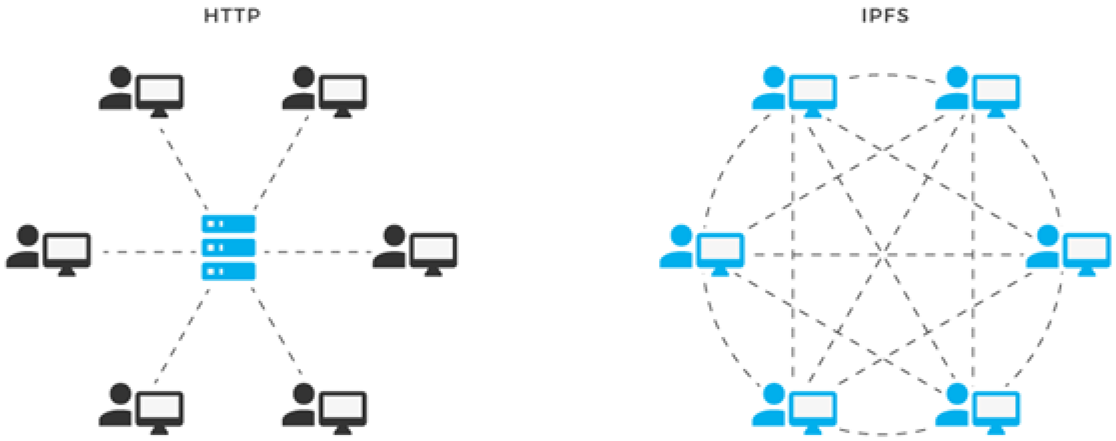

Figure 20: HTTP vs IPFS

14.2 Peer-to-peer storage & distribution

The IPFS protocol is a collection of protocols served from a swarm of IPFS nodes. Network IPFS nodes communicate regularly with hundreds of other nodes in the network, potentially across the wide internet. The IPFS network stack features:

- Transport: IPFS can use any transport protocol, and is best suited for WebRTC DataChannels (for browser connectivity) or uTP.
- Reliability: IPFS can provide reliability if underlying networks do not provide it, using uTP or SCTP.
- Connectivity: IPFS also uses the ICE NAT traversal techniques.
- Integrity: optionally checks integrity of messages using a hash checksum.
- Authenticity: optionally checks authenticity of messages using HMAC with sender's public key, see (Benet 2017).

Moreover, on top of the network, IPFS achieves reliable routing and data exchange through its DHT & BitSwap protocol, which enables IPFS to form massive peer-to-peer system for storing and distributing blocks quickly and robustly.

Moreover, on top of the network, IPFS achieves reliable routing and data exchange through its DHT & BitSwap protocol, which enables IPFS to form massive peer-to-peer system for storing and distributing blocks quickly and robustly.

14.3 Object Merkle DAG

On top of the efficient and robust peer-to-peer network, IPFS builds a Merkle DAG, a directed acyclic graph where links between objects are cryptographic hashes of the targets embedded in the sources. This is a generalization of the Git data structure. Merkle DAGs provide IPFS many useful properties, including:

1. Content Addressing: all content is uniquely identified by its multi-hash checksum, including links.
2. Tamper resistance: all content is verified with its checksum. If data is tampered with or corrupted, IPFS detects it.
3. Deduplication: all objects that hold the exact same content are equal, and only stored once. This is particularly useful with index objects, such as git trees and commits, or common portions of data, see (Benet 2017).

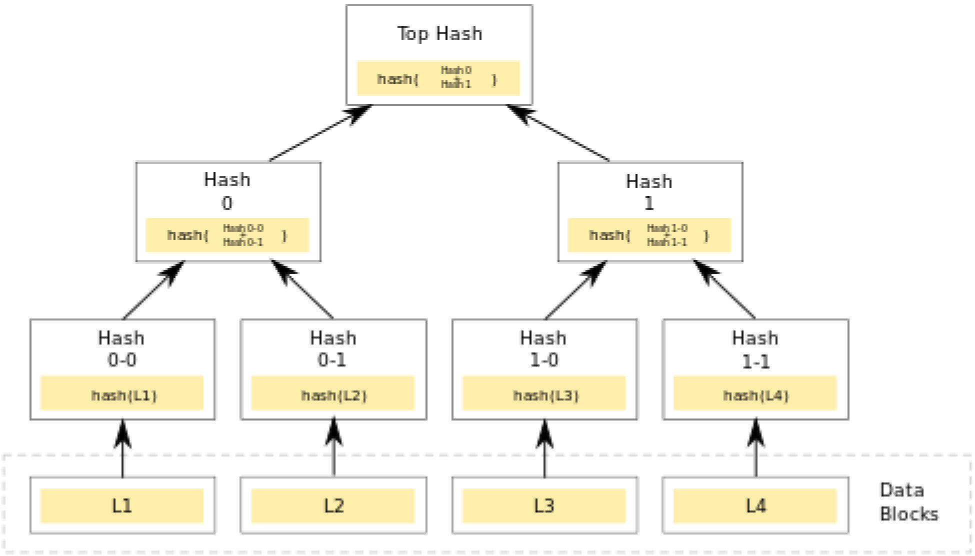

Figure 21: Merkle DAG

With the corruption-proof and duplication-free Merkle DAG structure, IPFS is able to build a reliable and scalable data storage system on top of the peer-to-peer network

14.4 CromoLab and IPFS

The working of Cromo Lab relies on big volumes of data, including terabytes of historical cryptocurrency market data that continue to accumulate every second, as well as the large amount of prediction results produced by the algorithm. Moreover, integrity and constant availability of the data are critical to the performance of the service. Therefore, Cromo Lab needs a scalable and reliable data storage and distribution solution that can efficiently handle large data throughput whilst guaranteeing the integrity of the data. This is where IPFS comes in.

IPFS is ideal for Cromo Lab's data storage and distribution needs, for its following merits:

1. No central point of failure - permanent and reliable availability One of the major risks of using central database for storage and distribution is the existence of a central point of failure. Failure of a central database can lead to severe damages, from temporary service

downtime to permanent loss of important data. The GitLab database outage this January, see https://about.gitlab.com/2017/02/01/gitlab-dot-com-database-incident/, which resulted in significant loss of production data, as well as the Amazon AWS outage in February, which paralyzed a large number of websites and services including Quoro, Trello and Wix, are just two of the many incidents that highlight the fragility of services relying on central databases. IPFS, on the other hand, has no central point of failure. As data is stored on the peer-to-peer network across numerous devices, failure of one or several device does not affect overall availability of the data. The decentralized nature of IPFS ensures constant and permanent data availability.

2. Content addressing with cryptographic hashes - corruption-proof Cromo Lab's algorithm relies on accurate data to generate accurate predictions. Therefore, integrity of the data stored is essential to the performance of our service. Traditional central databases are prone to data corruption resulting from software or hardware failure. On the other hand, with IPFS, all data is addressed and verified with its cryptographic hash. The system automatically detects when data is tampered or corrupted, therefore ensuring data integrity.
3. Distributed storage - scalable storage & high-performace distribution As the amount of market data and prediction results continues to grow, Cromo Lab's storage capacity needs to scale accordingly. With its highly flexible network protocol, IPFS is capable of heterogeneous scalability, able to readily expand its capacity by distributing storage needs to a wide network of different resources. Moreover, in terms of data distribution, the peer-to-peer network has significant advantage against traditional centralized databases. Compared to the traditional approach where data is distributed from a single server, IPFS distributes pieces of data from multiple devices simultaneously, and thus improves speed performance and saves bandwidth costs by up to 60%.

14.5 Storage solutions

At Cromo Lab, we aim to engineer a robust and efficient data storage and access system on top of IPFS. For different types of data with different access needs, we designed different data structures to suit the needs.

1. Archival data:

Archival data that is not frequently accessed or altered (e.g. historical market data) will be stored as single files into the IPFS swarm.

2. Updated data:

For data that will be constantly updated (e.g. update-to-date market data & prediction results), a different data structure is needed. Such data will be stored in a hash linked list structure. Specifically, data will be stored in blocks, with each block containing data from a certain timeframe, along with an IPFS hash pointing to the previous block. When data is updated, a new block containing the most recent data as well as the IPFS hash of the latest block will be added to IPFS. Additionally, the hash of the most recent block (pointer to the root node) will be kept and updated. By starting from most recent block and tracing the IPFS hashes, one can traverse the linked list and retrieve all the data. This design is analogous to the structure of blockchain. It enables frequent updates of data, whilst avoiding storage redundancy and repeated reading and writing into one large single file.

3. Frequent access of different time frames

In most cases, only the most recent data will be frequently accessed. In such case, the aforementioned hash linked list structure would be sufficient. However, in case where there is frequent needs for accessing data from various time ranges, the linked list structure is inefficient as it requires linear ($O(n)$) time to locate the corresponding data blocks. Instead, we improve the search efficiency by keeping an additional hash list, which contains IPFS hashes of the data blocks along with the corresponding time frames of the data. As the hash list will be sorted by time, more efficient binary search can be used to improve the search time to logarithmic ($O(\log(n))$) order.

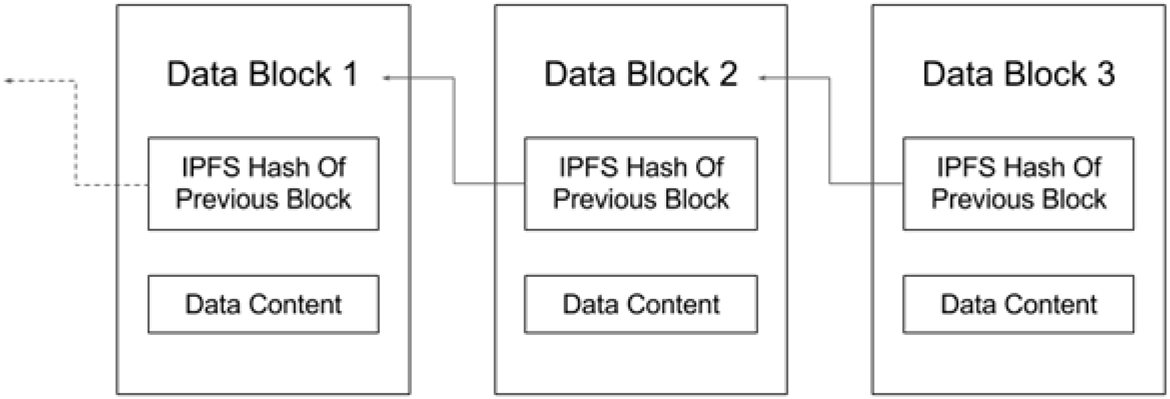
Data Block 1
IPFS Hash Of Previous Block
Data Content
Data Block 2
IPFS Hash Of Previous Block
Data Content
Data Block 3
IPFS Hash Of Previous Block
Data Content

Figure 22: IPFS block updating

14.6 Incentivize data hosting on IPFS

In order to ensure accessibility and improve distribution performance, Cromo Lab will also provide incentivize IPFS nodes to host the data. We have two potential solutions:

1. Token reward

The team may allocate a pool of tokens for storage rewards. We incentivize community members to run IPFS nodes and host our data with token rewards proportionate to the amount of storage and accessibility provided.

2. FileCoin

Filecoin is a data storage incentivization solution currently under development.It is a decentralized storage network than can run on top of IPFS, where miners receive token rewards for providing storage and distribution to clients, see (Labs 2017). If the Filecoin solution proves to be robust and cost-effective, in the long run Cromo Lab may also utilize the Filecoin network to harness to storage resources of a broader network.

References

Alexander, C., and A. Dimitriu. 2005a. “Indexing and Statistical Arbitrage: Tracking Error or Cointegration?” *Journal of Portfolio Management* 31: 50–63.

———. 2005b. “Indexing, Cointegration and Equity Market Regimes.” *International Journal of Finance and Economics* 10: 213–31.

Ardia, D., Gatarek L.T., L. Hoogerheide, and H.K. Van Dijk. 2016. “Return and Risk of Pairs Trading Using a Simulation-Based Bayesian Procedure for Predicting Stable Ratios of Stock Prices.” *Econometrics* 4 (1).

Ardia, D., L. Hoogerheide, and L.T. Gatarek. 2017. “A New Bootstrap Test for Multiple Assets Joint Risk Testing.” *Journal of Risk* 19 (4).

Benet, J. 2017. “IPFS - Content Addressed, Versioned, P2P File System.” https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg

Chu, J., S. Nadarajah, and S. Chan. 2015. “Statistical Analysis of the Exchange Rate of Bitcoin.” *PloS One*, 1–27.

Duan, J-Ch., and S.R. Pliska. 2004. “Option Valuation with Co-Integrated Asset Prices.” *Journal of Economic Dynamics and Control* 24 (4): 727–54.

Durbin, J., and S.J. Koopman. 2012. *Time Series Analysis by State Space Methods.* Oxford University Press, (2. ed.).

Feller, W. 1957. *An Introduction to Probability Theory and Its Applications (Volume 1).* John Wiley & Sons Inc., (2. ed.).

Gatev, E., W. N. Goetzmann, and K. G. Rouwenhorst. 2006. “Pairs Trading: Performance of a Relative-Value Arbitrage Rule.” *The Review of Financial Studies* 19: 797–827.

Gillespie, T., and C. Ulph. 2001. “Pair Trades Methodology: A Question of Mean Reversion.” Proceedings of International Conference on Statistics, Combinatorics and Related Areas and the 8th International Conference of Forum for Interdisciplinary Mathematics, NSW.

Grammig, Melvin, J., and C. Schlag. 2005. “Internationally cross-listed stock prices during overlapping trading hours: price discovery and exchange rate effects.” *Journal of Financial Econometrics* 12: 139–64.

Grassi, S., and L. Catania. 2017. “Modelling Crypto-Currencies Financial Time-Series.” https://ssrn.com/abstract=3028486.

Gutierrez, J. A., and Y. Tse. 2011. “Illuminating the Profitability of Pairs Trading: A Test of the Relative Pricing Efficiency of Markets for Water Utility Stocks.” *The Journal of Trading* 6 (2): 50–64.

Hasbrouck, J. 1988. “One security, many markets: determining the contributions to price discovery.” *The Journal of Finance* 50: 1175–99.

Hencic, A., and C. Gourieroux. 2014. “Noncausal Autoregressive Model in Application to Bitcoin USD Exchange Rate.” Proceedings of the 7th Financial Risks International Forum, 125–25.

Johansen, S. 1988. “Statistical analysis of cointegration vectors.” *Journal of Economic Dynamics and Control* 12: 231–54.

———. 2006. *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models.* Oxford

University Press, Oxford (2. ed.).

Johansen, S., and L. Gatarek. 2017. “The Role of Cointegration for Optimal Hedging with Heteroscedastic Error Term.” *Journal of Econometrics* final revision. http://pure.au.dk/portal/en/publications/id(c0d43
8cba-45fe-b554-f0b716c4d389).html.

Jong, F. de, and P.C. Schotman. 2010. “Price discovery in fragmented markets.” *Journal of Financial Econometrics* 8: 1–28.

Juhl, T., I. Kawaller, and P. Koch. 2012. “The Effect of the Hedge Horizon on Optimal Hedge Size and Effectiveness When Prices Are Cointegrated.” *Journal of Futures Markets* 32 (9): 837–76.

Kasa, K. 1992. “Common stochastic trends in international stock markets.” *Journal of Monetary Economics* 29: 95–124.

Labs, Protocol. 2017. “Filecoin: A Decentralized Storage Network.” https://filecoin.io/filecoin.pdf.

Lehmann, B.N. 2002. “Some desiderata for the measurement of price discovery across markets.” *Journal of Financial Markets* 5: 259–76.

Lévy, P. 1939. “Sur Certains Processus Stochastiques Homogènes.” *Compositio Mathematica* 7: 283–339.

Lin, Y.-X., M. McCrae, and C. Gulati. 2006. “Loss Protection in Pairs Trading Through Minimum Profit Bounds: A Cointegration Approach.” *Journal of Applied Mathematics and Decision Sciences*, 1–14.

Mörters, P., and Y. Peres. 2010. *Brownian Motion*. Cambridge University Press, Cambridge.

Sapuric, S., and A. Kokkinaki. 2014. “Bitcoin Is Volatile. Isnt That Right.” Business Information Systems Workshops, 255–65.

Shephard, N., and M.K. Pitt. 1997. “Likelihood Analysis of Non-Gaussian Measurement Time Series.” *Biometrika* 84: 653–67.

———. 1999. “Filtering via Simulation: Auxiliary Particle Filters.” *Journal of the American Statistical Association* 94 (446): 590–99.

Van Dijk, H., and R. Kleijn. 2006. “Bayes Model Averaging of Cyclical Decompositions in Economic Time Series.” *Journal of Applied Econometrics* 21 (2). http://dx.doi.org/10.1002/jae.823: 191–212.

Van Dijk, H., and T. Kloek. 1978. “Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo.” *Econometrica* 46 (1).

Van Dijk, H., A.C. Harvey, and T.M. Trimbur. 2007. “Trends and Cycles in Economic Time Series: A Bayesian Approach.” *Journal of Econometrics* 140 (2). http://dx.doi.org/10.1016/j.jeconom.2006.07.006: 618–49.

Van Dijk, H., Billio M., Casarin R., and Ravazzolo F. 2013. “Time-Varying Combinations of Predictive Densities Using Nonlinear Filtering.” *Journal of Econometrics* 177 (2).

Vidyamurthy, G. 2004. *Pairs Trading: Quantitative Methods and Analysis*. New York: Wiley.