The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy

David Garcia, Claudio J. Tessone, Pavlin Mavrodiev, Nicolas Perony

S1 Electronic Supplementary Material

S1.1 Reconstructing Google search time series

We retrieved Google trends data for the term “bitcoin” on November 5th, 2013, using the csv export function[1]. This service provides weekly rescaled search volumes since 2004, but daily search volumes can only be downloaded for time intervals up to three months. We retrieved the time series of daily search volumes for the period since the beginning of 2010 as follows:

- First, we do one query for the weekly rescaled volumes since Jan 1st 2010, represented as the blue dots in Fig. S1.
- Second, we perform a set of queries for a rolling time window of two months, returning daily volumes. For each week in the whole time period, we have the rescaled daily volumes, shown as the red inset in Fig. S1.
- Third, we add the daily resolution to the weekly volumes by rescaling the intra-week volumes of the second step, approximating this way the whole time series of daily search volumes.

To validate the correctness of this reconstruction method, we applied it to simulated time series of Brownian motion, white and pink noise. For each simulated time series, we create a weekly volume version, and a set of intra-week volumes as we got from Google trends. We use that data to reconstruct the simulated time series, and evaluate the error introduced by the reconstruction method comparing the initial and the reconstructed time series. Fig. S2 shows a simulation of Brownian motion, and its reconstructed values. The inset shows the scatter plot of both simulated and reconstructed values, showing that the reconstruction method provides a very precise, rescaled approximation of the original time series. This way, less than 0.2% of the variance was lost for Brownian motion, and less than 0.05% was lost for white and pink noise.

[1] `http://www.google.com/trends/explore`

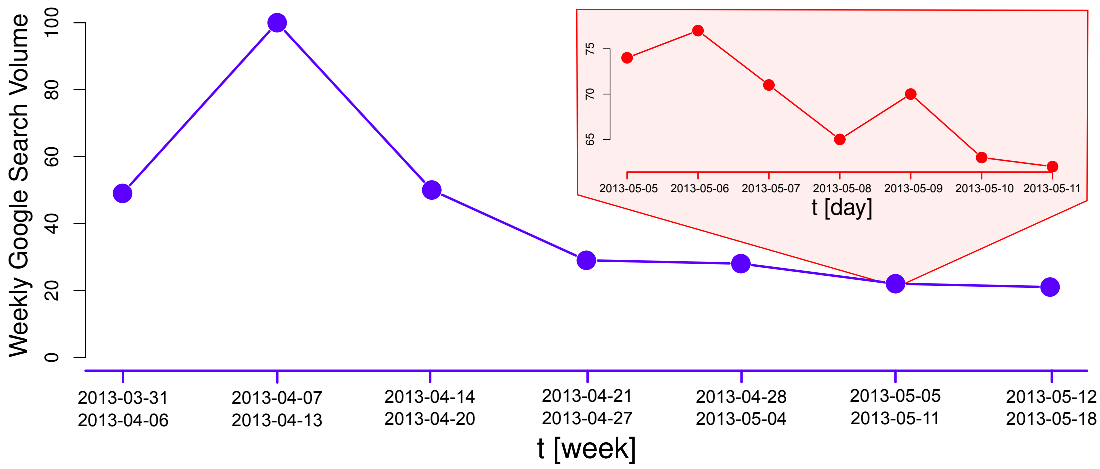

Figure S1: Example of Google Search Volume data. Search volume from Google trends can be extracted in two formats: weekly normalised volume (blue), and daily normalised volume (red, inset). Weekly volume data spans the whole analysis period, daily data can be retrieved for periods up to three months. Our reconstruction method combines these two sources to produce a daily time series for the whole period.

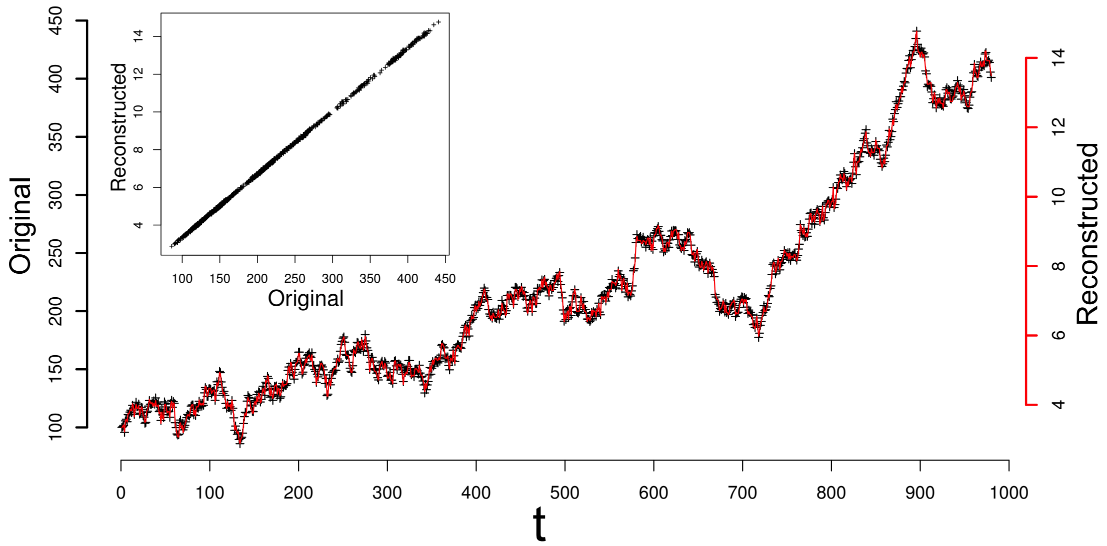

Figure S2: Results of reconstruction method. A simulated time series of Brownian motion (black dots), and its reconstructed version (red lines). Inset: scatter plot of simulated daily values and reconstructed ones. The reconstruction method loses less than 0.2% of the variance of the original time series. Additional tests with white and pink noise lose less than 0.05% of the variance.

S1.2 Reconstructing the number of Bitcoin users from the block chain dataset

A Bitcoin address is a unique 27-34 alphanumeric identifier of the following form – `31uEbMgunupShBVTewXjtqbBv5MndwfXhb`. It represents a possible destination for a Bitcoin payment. A Bitcoin user can have multiple addresses. To extract the addresses used for transacting in the Bitcoin network and to later identify unique users, we analysed the Bitcoin block chain up to November 2013. The block chain is a growing historical "book of records" containing full information about all transactions that take place in the network. Every Bitcoin client keeps a complete copy of the block chain, stored locally in the form of raw binary data. Once a user A decides to send a certain amount of BTC to user B, the client creates a transaction is listing (i) the desired amount to transfer, (ii) all of user A's addresses from which BTCs, adding up at least to the desired amount, are to be collected (the input field) and (iii) the addresses to which the desired amount should be sent (the output field). This transaction is then broadcasted to the network, and eventually written into the last block of the block chain. Since the block chain contains information about addresses only, we use the following two rules to match some of these addresses to individual users:

1. **Merging multiple input addresses.** As mentioned above, the input field of a transaction contains all addresses whose total BTC content must be combined for the transfer. To access the funds in a Bitcoin address, a user must possess a secret number, known as a "private key", which implies that every Bitcoin address has a unique owner. It follows, then, that all input addresses must belong to the same user – the one who knows the private key[2]. Hence, for each transaction we merge the input addresses and assign them to a unique user. Moreover, if a transaction has an input address which has already been assigned then the remaining input addresses are also assigned to the same user.

2. **Identifying change addresses** Due to the design of the Bitcoin protocol, the content of a Bitcoin address must be fully spent, once this address has been chosen to be an input for a transaction. Consequently, the total amount of BTC contained in the transaction input must be fully spent. Often in practice, the total funds in the inputs are in excess of the desired transfer amount. In such case, the remaining Bitcoins must be returned to the sender as "change". In the vast majority of current use, the Bitcoin client accomplishes this by creating internally a new valid Bitcoin address, owned by the sender, to which the change is to be transferred. The funds collected in this change address can be reclaimed by the Bitcoin client for the next transaction of the sender. Moreover, the process is invisible to the user – the change address is not known, unless the user manually inspects the output

[2]It is highly unlikely that multiple users control the different input addresses, as this implies that they have shared the private keys among them

field of the transation in the block chain. Therefore, it is unlikely that the user will provide this address as recipient for a future transaction, which implies that the change address appears only once in the output field of a transaction. This is our second rule – we scan the output field of each transaction and look for an address that has only been used once as output in the block chain. If there is only one such address in a transaction, we assign it to the sender. In case of multiple addresses meeting this criterion, we conservatively do not label a change address for this transaction.

S2 Time series stationarity

X	Data	ADF	KPSS	$A(1)$	ΔX	ADF	KPSS	$A(1)$
W	Twitter	$< 10^{-30}$	< 0.01	0.816	ΔW	$< 10^{-57}$	> 0.1	-0.299
W	Facebook	$< 10^{-57}$	< 0.01	0.302	ΔW	$< 10^{-57}$	> 0.1	-0.372
S	Google	$< 10^{-14}$	< 0.01	0.999	ΔS	$< 10^{-57}$	> 0.1	0.277
S	Wikipedia	$< 10^{-20}$	< 0.01	0.929	ΔS	$< 10^{-57}$	> 0.1	0.145
U	Client Downloads	$< 10^{-10}$	< 0.01	0.990	ΔU	$< 10^{-57}$	> 0.1	-0.036
U	Blockchain Users	$< 10^{-9}$	< 0.01	0.999	ΔU	$< 10^{-57}$	> 0.1	-0.033
U	Addresses	$< 10^{-12}$	< 0.01	0.999	ΔU	$< 10^{-57}$	> 0.1	-0.164
P	MtGox	0.92	< 0.01	0.999	ΔP	$< 10^{-57}$	0.0822	0.178
P	BTCChina	0.24	< 0.01	0.998	ΔP	$< 10^{-57}$	> 0.1	0.326
P	MtGox EUR	0.38	< 0.01	0.998	ΔP	$< 10^{-49}$	> 0.1	0.216
P	BTC-de	0.25	< 0.01	0.998	ΔP	$< 10^{-57}$	> 0.1	0.23

Table S1: Results of stationarity tests for each variable X and their first difference ΔX.

S3 Lagged correlation analysis

We study the feedback dynamics between these four variables by means of a lagged correlation analysis. Fig. S3 shows the result of lagged Pearson's cross-correlation tests ($\rho(X(t), Y(t + \delta t))$ between all pairs of our four variables. We observe that changes in the exchange rate of USD to BTC on Mt. Gox (hereafter price) lead to corresponding changes in search volume, with ρ peaking at one day (fast response). However, increases in search volume lead to decreases in price after a few days. Increases in price lead to an increased number of client downloads after 1-2 days. Search volume precedes word-of-mouth levels (tweet ratio) by one day, showing that information search precedes information sharing. There are faint positive correlations between client downloads and word of mouth, as well as between word of mouth and price, but the lagged correlation analysis is not sufficient to conclude dependencies between these two variables, which motivates our vector autoregression analysis. All the relations reported here are consistent when using Wikipedia views instead of Google searches as the variable for S_t. These two ways of measuring search interest are strongly correlated at lag $\delta = 0$. While such an analysis gives useful insights into the dynamics of the strict pairwise correlations between the variables of our study, it does not account for coupled correlations between multiple variables, for example between search and users on price. Our VAR analysis takes into account the multidimensional nature of our analysis, and reveals stronger relations that were not observable in this pairwise analysis.

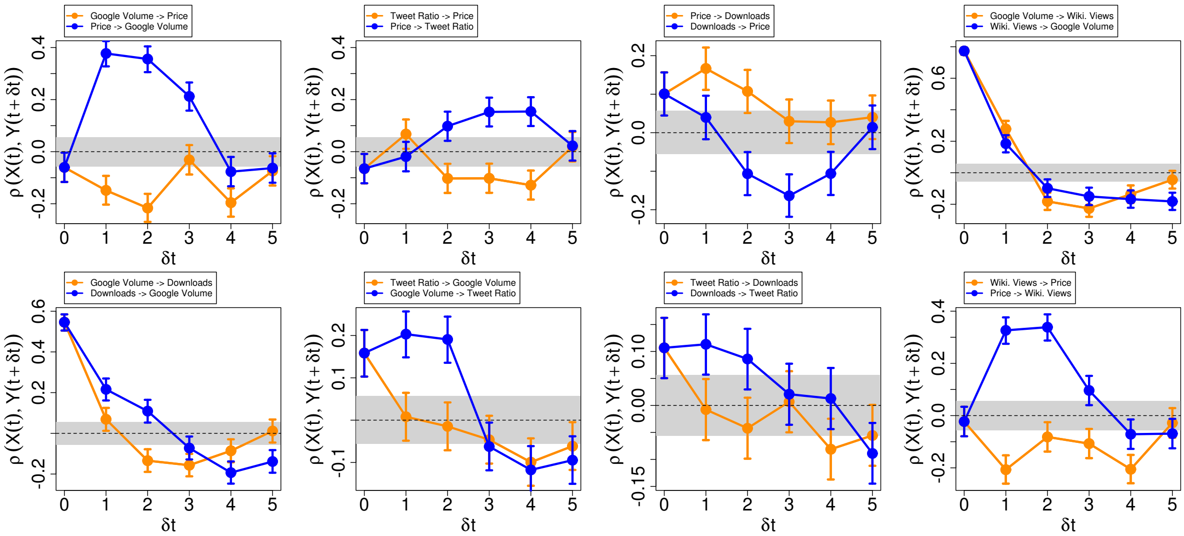

Figure S3: Cross-correlation function between pairs of variables in our analysis. Dots indicate Pearson's correlation coefficient between lagged variables, error bars show 95% confidence intervals. The greyed area around 0 shows the average confidence interval of the correlation for 1000 permutations of the empirical data.

S4 VAR Results without normalisation

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.153** $(7.2 * 10^{-08})$	**0.00038** $(5.3 * 10^{-05})$	**0.027** $(4.4 * 10^{-04})$	**−1.867** $(2.3 * 10^{-11})$
ΔU_t	**67.166** $(1.6 * 10^{-10})$	**−0.143** $(3.0 * 10^{-05})$	−0.790 $(7.8 * 10^{-01})$	**462.059** $(5.8 * 10^{-06})$
ΔW_t	−0.117 $(2.0 * 10^{-01})$	8.186 $(9.9 * 10^{-01})$	**−0.320** $(9.4 * 10^{-34})$	**7.070** $(1.3 * 10^{-14})$
ΔS_t	**0.048** $(1.5 * 10^{-46})$	5.700 $(5.9 * 10^{-01})$	−0.000 $(5.9 * 10^{-01})$	**0.293** $(5.0 * 10^{-20})$

Table S2: Vector Auto-Regression results for P_t as price on Mt. Gox, W_t as tweet ratio, S_t as Google search volume, and U_t as number of client downloads, without renormalisation.

S5 VAR results with variable replacements

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.158** $(1.3 * 10^{-08})$	**0.148** $(3.1 * 10^{-06})$	**0.094** $(7.4 * 10^{-04})$	**−0.291** $(2.5 * 10^{-19})$
ΔU_t	**0.176** $(7.9 * 10^{-10})$	**−0.120** $(2.1 * 10^{-04})$	−0.000 $(9.8 * 10^{-01})$	**0.134** $(4.3 * 10^{-05})$
ΔW_t	−0.043 $(8.8 * 10^{-02})$	0.026 $(3.5 * 10^{-01})$	**−0.309** $(5.4 * 10^{-32})$	**0.221** $(7.5 * 10^{-14})$
ΔS_t	**0.316** $(1.5 * 10^{-30})$	0.054 $(7.3 * 10^{-02})$	−0.007 $(7.9 * 10^{-01})$	**0.126** $(4.3 * 10^{-05})$

Vector Auto-Regression results for P_t as price on Mt. Gox, W_t as tweet ratio, S_t as Wikipedia views, and U_t as number of client downloads.

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.162** $(2.0 * 10^{-08})$	**0.151** $(1.0 * 10^{-05})$	**0.086** $(2.9 * 10^{-03})$	**−0.232** $(3.0 * 10^{-11})$
ΔU_t	**0.194** $(2.7 * 10^{-11})$	**−0.136** $(7.6 * 10^{-05})$	0.055 $(5.8 * 10^{-02})$	**0.149** $(1.9 * 10^{-05})$
ΔW_t	**−0.148** $(1.4 * 10^{-08})$	**−0.147** $(1.7 * 10^{-06})$	**−0.446** $(6.0 * 10^{-59})$	**0.303** $(1.8 * 10^{-21})$
ΔS_t	**0.373** $(5.7 * 10^{-43})$	0.006 $(8.4 * 10^{-01})$	**−0.079** $(2.6 * 10^{-03})$	**0.303** $(2.6 * 10^{-21})$

Vector Auto-Regression results for P_t as price on Mt. Gox, W_t as Facebook reshares, S_t as Google search volume, and U_t as number of client downloads.

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.161** $(1.4 * 10^{-08})$	**0.089** $(3.7 * 10^{-03})$	**0.096** $(7.1 * 10^{-04})$	**−0.172** $(7.1 * 10^{-09})$
ΔU_t	**0.143** $(1.1 * 10^{-07})$	**−0.112** $(1.2 * 10^{-04})$	−0.028 $(2.9 * 10^{-01})$	**0.110** $(8.9 * 10^{-05})$
ΔW_t	−0.033 $(1.9 * 10^{-01})$	0.007 $(7.9 * 10^{-01})$	**−0.320** $(1.0 * 10^{-33})$	**0.242** $(3.8 * 10^{-19})$
ΔS_t	**0.384** $(1.7 * 10^{-46})$	0.040 $(1.4 * 10^{-01})$	−0.016 $(5.3 * 10^{-01})$	**0.295** $(5.7 * 10^{-27})$

Vector Auto-Regression results for P_t as price on Mt. Gox, W_t as tweet ratio, S_t as Google search volume, and U_t as new number of users detected in the blockchain.

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.167** $(5.0 * 10^{-09})$	0.040 $(1.8 * 10^{-01})$	**0.099** $(5.3 * 10^{-04})$	**−0.163** $(4.1 * 10^{-08})$
ΔU_t	**0.129** $(1.5 * 10^{-06})$	**−0.235** $(4.3 * 10^{-16})$	−0.030 $(2.5 * 10^{-01})$	**0.108** $(1.1 * 10^{-04})$
ΔW_t	−0.034 $(1.7 * 10^{-01})$	0.017 $(5.1 * 10^{-01})$	**−0.321** $(7.6 * 10^{-34})$	**0.240** $(4.1 * 10^{-19})$
ΔS_t	**0.384** $(1.4 * 10^{-46})$	0.038 $(1.6 * 10^{-01})$	−0.016 $(5.2 * 10^{-01})$	**0.297** $(1.9 * 10^{-27})$

Vector Auto-Regression results for P_t as price on Mt. Gox, W_t as tweet ratio, S_t as Google search volume, and U_t as new number of addresses in the blockchain.

Table S3: Vector Auto-Regression results with alternative metrics of the variables U, W, and S.

S6 VAR results for other markets and currencies

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.188** $(3.9 * 10^{-08})$	**0.163** $(5.1 * 10^{-05})$	**0.249** $(4.8 * 10^{-12})$	**−0.356** $(2.4 * 10^{-16})$
ΔU_t	**0.222** $(5.2 * 10^{-10})$	−0.057 $(1.7 * 10^{-01})$	−0.037 $(3.1 * 10^{-01})$	**0.205** $(4.3 * 10^{-06})$
ΔW_t	**−0.103** $(2.0 * 10^{-03})$	−0.004 $(9.0 * 10^{-01})$	**−0.116** $(8.5 * 10^{-04})$	**0.398** $(1.9 * 10^{-20})$
ΔS_t	**0.404** $(2.2 * 10^{-35})$	**0.100** $(6.0 * 10^{-03})$	−0.060 $(6.2 * 10^{-02})$	**0.315** $(1.7 * 10^{-15})$

Vector Auto-Regression results for P_t as price on Mt. Gox in EUR, W_t as tweet ratio, S_t as Google search volume, and U_t as number of client downloads.

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.306** $(2.5 * 10^{-21})$	**0.170** $(7.2 * 10^{-06})$	**0.082** $(9.6 * 10^{-03})$	**−0.352** $(1.8 * 10^{-19})$
ΔU_t	**0.249** $(2.0 * 10^{-13})$	**−0.093** $(2.0 * 10^{-02})$	−0.018 $(5.7 * 10^{-01})$	**0.164** $(5.4 * 10^{-05})$
ΔW_t	**−0.121** $(1.6 * 10^{-04})$	0.052 $(1.7 * 10^{-01})$	**−0.275** $(5.3 * 10^{-17})$	**0.247** $(2.7 * 10^{-10})$
ΔS_t	**0.478** $(1.6 * 10^{-54})$	0.046 $(1.7 * 10^{-01})$	−0.045 $(1.1 * 10^{-01})$	**0.252** $(6.5 * 10^{-13})$

Vector Auto-Regression results for P_t as price on BTC China, W_t as tweet ratio, S_t as Google search volume, and U_t as number of client downloads.

	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.169** $(1.5 * 10^{-06})$	**0.149** $(3.1 * 10^{-04})$	**0.139** $(1.1 * 10^{-04})$	**−0.353** $(1.7 * 10^{-15})$
ΔU_t	**0.259** $(8.8 * 10^{-13})$	**−0.087** $(3.8 * 10^{-02})$	**−0.090** $(1.4 * 10^{-02})$	**0.235** $(1.6 * 10^{-07})$
ΔW_t	**−0.184** $(4.7 * 10^{-08})$	0.027 $(4.7 * 10^{-01})$	**−0.083** $(1.5 * 10^{-02})$	**0.366** $(1.0 * 10^{-17})$
ΔS_t	**0.439** $(1.2 * 10^{-40})$	0.056 $(1.2 * 10^{-01})$	**−0.151** $(2.9 * 10^{-06})$	**0.360** $(1.3 * 10^{-19})$

Vector Auto-Regression results for P_t as price on BTC-de, W_t as tweet ratio, S_t as Google search volume, and U_t as number of client downloads.

Table S4: Vector Auto-Regression results with alternative exchange markets and currencies to measure P.

S6.1 VAR applied over different periods

P1,P2	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.317** $(5.5 * 10^{-21})$	**−0.150** $(2.1 * 10^{-03})$	−0.016 $(6.1 * 10^{-01})$	**0.146** $(2.3 * 10^{-03})$
ΔU_t	−0.005 $(8.6 * 10^{-01})$	**−0.169** $(5.9 * 10^{-04})$	0.005 $(8.7 * 10^{-01})$	**−0.118** $(1.5 * 10^{-02})$
ΔW_t	−0.011 $(7.0 * 10^{-01})$	0.006 $(8.8 * 10^{-01})$	**−0.372** $(6.1 * 10^{-34})$	**0.085** $(4.9 * 10^{-02})$
ΔS_t	**−0.099** $(2.5 * 10^{-03})$	**0.122** $(1.2 * 10^{-02})$	0.015 $(6.2 * 10^{-01})$	**−0.377** $(1.2 * 10^{-14})$

P3	ΔP_{t-1}	ΔU_{t-1}	ΔW_{t-1}	ΔS_{t-1}
ΔP_t	**0.176** $(1.0 * 10^{-03})$	**0.163** $(9.7 * 10^{-03})$	**0.283** $(1.0 * 10^{-06})$	**−0.329** $(1.6 * 10^{-06})$
ΔU_t	**0.220** $(7.9 * 10^{-05})$	−0.030 $(6.4 * 10^{-01})$	−0.023 $(6.8 * 10^{-01})$	**0.188** $(7.1 * 10^{-03})$
ΔW_t	−0.031 $(5.4 * 10^{-01})$	−0.007 $(9.0 * 10^{-01})$	−0.063 $(2.4 * 10^{-01})$	**0.431** $(1.0 * 10^{-10})$
ΔS_t	**0.382** $(2.9 * 10^{-14})$	**0.117** $(3.8 * 10^{-02})$	−0.038 $(4.5 * 10^{-01})$	**0.302** $(9.4 * 10^{-07})$

Table S5: Vector autoregression results for P_t as price on Mt. Gox, S_t as Google search volume, and U_t as identified users in the block chain, for the period between the opening of Mt. Gox and the end of the second bubble (P1+P2), and the third bubble (P3).

S6.2 Extended VAR results

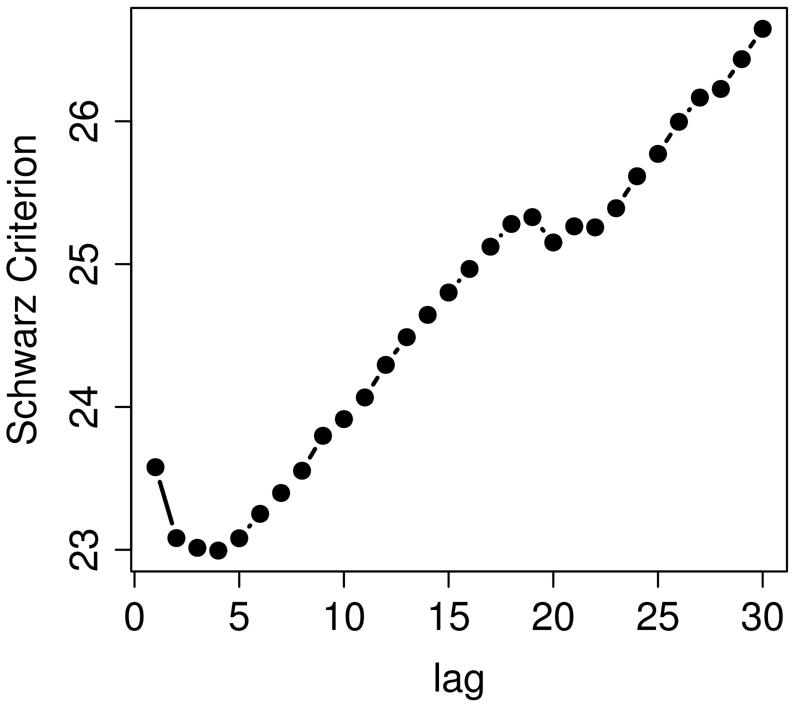

Figure S4: Schwarz Bayesian information criterion over VAR of lags from 1 to 30. The Schwarz criterion reaches its minumum at lag = 4 days.

Price increase	33.18	27.23	24.64	22.05	20.28	19.94	17.71	17.44	15.37	14.13
VAR estimate	14.88	12.41	12.26	17.56	1.18	23.95	4.13	10.17	1.46	6.30
Price decrease	-73.36	-30.02	-25.72	-21.74	-18.82	-16.90	-15.19	-14.56	-14.50	-12.80
VAR estimate	-48	3.54	-18.57	-21.05	-1.47	-0.49	-3.96	-7.38	-2.23	-6.41

Table S6: Top 10 strongest price increases and top 10 strongest price decreases between November 30th, 2012 and October 30th, 2013, and VAR estimate with lag 4. The VAR correctly estimates the sign of the top 10 increases and of 9 out of the top 10 decreases.

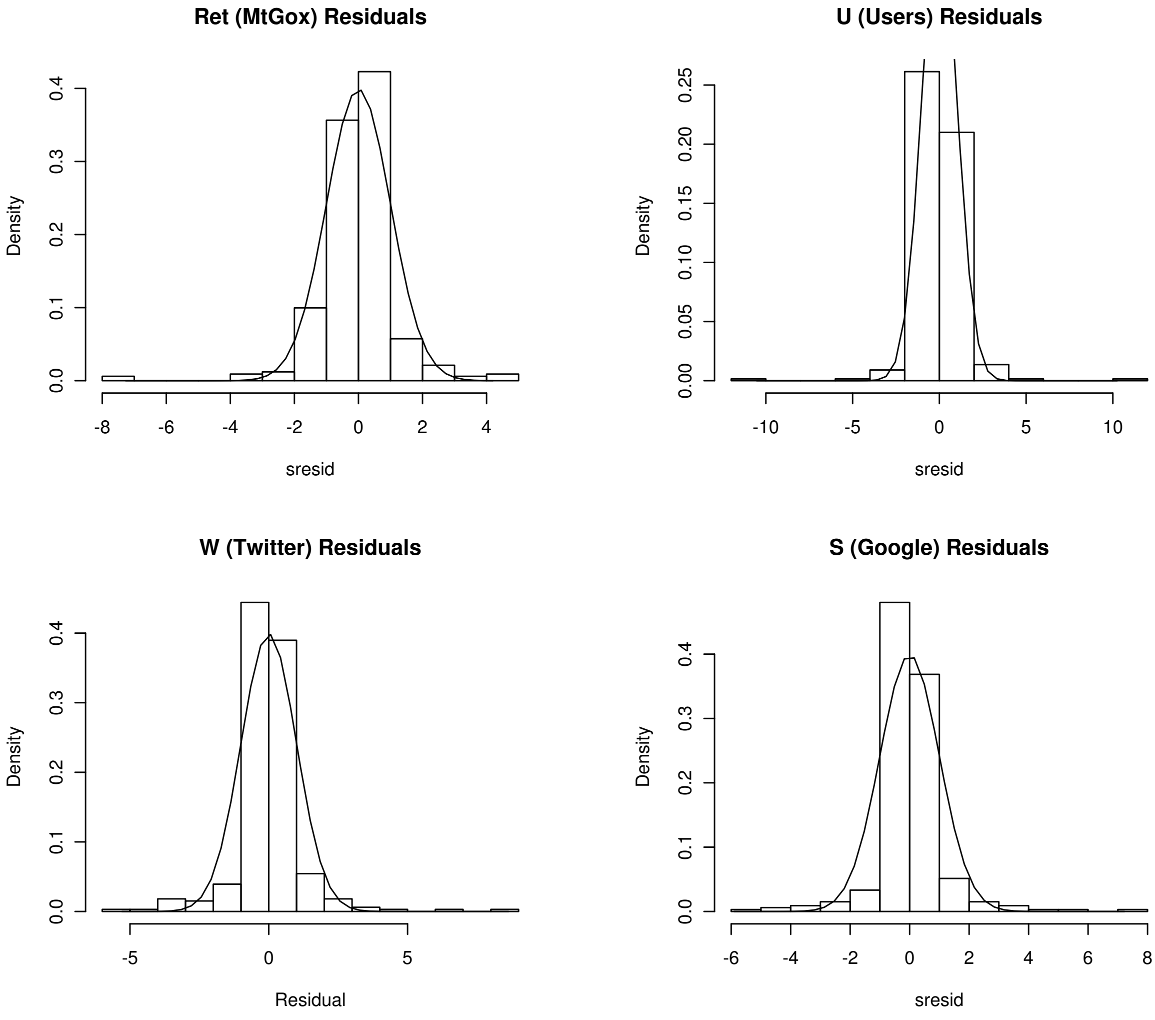

Figure S5: Distributions for the Studentised residuals of the four estimates of the model, which resemble normal distributions. Nevertheless, Shapiro-Wilk tests reject the normal distribution, indicating that improvements of the model are possible, especially for the outliers.

S7 Detailed results of impulse response functions

shock	P_{t+1}	P_{t+2}	S_{t+1}	S_{t+2}	W_{t+1}	W_{t+2}	U_{t+1}	U_{t+2}
P_t	0.94	0.14	–	-0.139	–	0.087	0.175	0.142
S_t	–	0.374	0.845	0.281	–	–	–	–
W_t	–	–	–	0.192	0.86	-0.277	–	–
U_t	–	0.175	0.517	–	–	–	0.818	-0.086

Table S7: Response function estimates of lags 1 and 2 under 97.5% confidence level (95% after Bonferroni correction).