Midterm Report Y'all

Thee (tmeensuk), Petros (pdawit), Dikshyant (drai), Kasemsan (kkongsal)

Introduction

In our blogpost #1, we outlined what we want to do, including the rough definitions of our models. We came up with two approaches: the Computational Power Approach and the Macroeconomics Approach. There, we only define our expected price functions and their arguments.

In this report, we process the data and visualize them so that we can construct our functions. We use the visualization to illustrate each of our implementations of our functions and figure out which of them make the most sense. Also, we take into account other details of our models such as limitations and restrictions, to capture the real world dataset.

Additionally, we discuss the data we are analyzing in the data section of this report. We discuss the questions we are investigating in our the Computational Power Approach and the Macroeconomics Approach sections.

Finally, we discuss our challenges and progress in the last section.

Data

As outlined in our last blogpost, since they provide the API which allows us to pull the data in the csv format, we mostly use the data from,

https://www.quandl.com/collections/markets/bitcoin-data

In the future reports, however, if we expand our models to cover other assets or precious metals--like S&P 500, gold price, or oil price--we will pull data from other sources as well. To do that, we might use Github open datasets.

The format of the API we are using to get the CSV files are as follow,

https://www.quandl.com/api/v3/datasets/<Group>/<DatasetName>.csv?api_key=<Key>

We obtain our API keys by creating free accounts with Qunadl.

For the Computational Power Approach, we use the following datasets: `BITSTAMPUSD-Bitcoin-Markets-bitstampUSD`, `Bitcoin-Market-Capitalization`, and `HRATE-Bitcoin-Hash-Rate`.

For the Macroeconomics Trend Approch, we use the following datasets: `BITSTAMPUSD-Bitcoin-Markets-bitstampUSD,EUR-EUR-BITCOIN-Weighted-Price,CNY-CNY-BITCOIN-Weighted-Price,BNC2-BNC-Digital-Currency-Indexed-EOD, MTGOXGBP-Bitcoin-Markets-mtgoxGBP`

For most datasets, we configure the API to get the data in the csv format the has date as the first column and value of the measurement as the second column.

Computational Power Approach

From our previous blogpost, we use $hashRate$ as the representation of the computational power and $marketCap$ being the the market capitalization of Bitcoin against the USD. We obtained the definition for the expected price as the function of $hashRate$ and $marketCap$ as follow,

$$E(price)_{i+k} = f(hashRate_i, marketCap_i)$$

Where i is the day we are considering, k is the number of days away from today that we are predicting in the future, $hashRate$ is the estimated number of giga hashes per second (billions of

hashes per second) the bitcoin network is performing, and $marketCap$ is the total number of bitcoins in circulation the market price in USD.
This equation is our hypothesis. We believe that this hypothesis works because the expected price (the value) of day $i + k$ should be the result of the productivity of day i .

However, this expected price does not capture the fluctuations of the price in the market as such fluctuations depend on interest rate risk, currency risk, operational risk, volatility risk, and other risks associated with the price that we might not know. Therefore, we formalized our model by taking that fluctuations as a factor we cannot measure, captured by this equation,

$$realPrice_i = E(price)_i + \varepsilon_i$$

Where i is the day we are considering, $realPrice$ is the price of bitcoin in one market against USD, and ε (epsilon) is the difference of the real price and the expected price. We do this because we think the expect price that we refer to is the real value of the bitcoin, while the $realPrice$ (or, the market price) is the price that the market values because of other factors (like risks) that we mention above.

Please note the our functions are calculated daily. For example, ε for today's price might not be the same as ε of yesterday's price.

We illustration our concept of the relationship of $realPrice$, $E(price)$, and ε below.

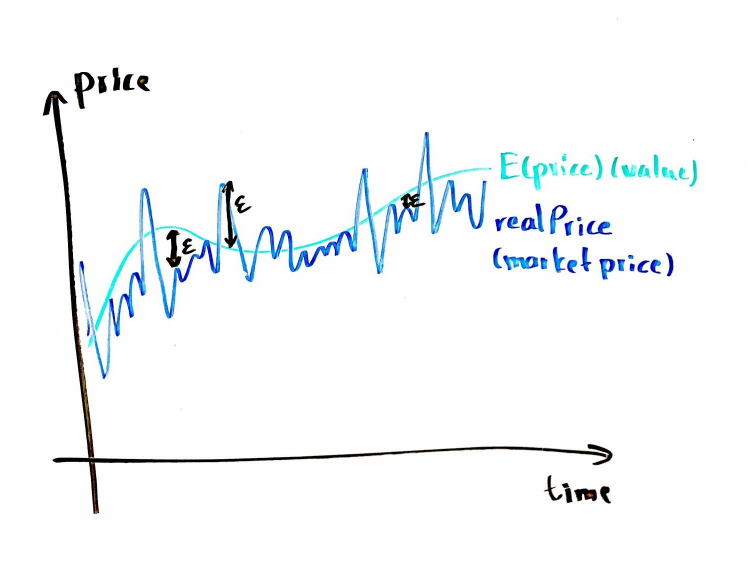


Figure 1: The relationship of $realPrice$, $E(price)$, and ε.

Now, we will work on how our prediction can reduce that ε we modeled. In other words, we need to smooth the curve so that we can eliminate the ε fluctuations. One way to do that is to use the Market Average, MA, for each day d, as defined below,

$$MA(length)_d = (\sum_{i=d-length}^{d} realPrice_i) / length$$

For example, today's value of $MA(50)$ is the average of the $realPrice$ from 50 days before today until today.

The next question we need to answer is that what argument we should supply to MA. Therefore, we illustrate $MA(10)$, $MA(50)$, and $MA(200)$ so that they might help us decide which one fits our model most.

In order to do that, we cleaned the data and compute our MA results with our Python code. We illustrate them using excel. The illustration is shown below.

The red line is the $realPrice$, the green line is $MA(10)$, the blue line is $MA(50)$, and the purple line is $MA(200)$. Since all those three make sense, we think we can use any of them to use with our model. For this visualization, we will use $MA(50)$ because it does not mimic the $realPrice$ too much, and it is not too smooth. (In reality, we can pick any of them depending how far in the future we want to predict, but they might yield the results differently.) We, then, form our hypothesis that MA of day i is the expected price of day i:

$$MA(k)_i = E(price)_i$$

Next, we construct the model for $E(price)$. In this case, we believe that two variable linear regression might work. We, therefore, model that following,

$$MA(k)_i = f(hashRate_{i-k}, marketCap_{i-k})$$
$$MA(k)_i = \alpha \cdot hashRate_{i-k} + \beta \cdot marketCap_{i-k} + \gamma$$

To test this hypothesis, we create a csv file that contains 4 columns: date(i), $MA(50)_i$, $hashRate_{i-50}$, and $marketCap_{i-50}$, and use Excel to calculate the two variable regression.

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.870000705							
R Square	0.756901226							
Adjusted R S	0.756484247							
Standard Err	99.93674816							
Observations	1169							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	2	36258131.4	18129065.7	1815.20214	0			
Residual	1166	11645254.3	9987.35363					
Total	1168	47903385.8						
	Coefficients	*Standard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	3530.089114	67.1830818	52.5443165	0	3398.27587	3661.90236	3398.27587	3661.90236
hash_rate	0.000526514	8.901E-06	59.1522557	0	0.00050905	0.00054398	0.00050905	0.00054398
total_cap	-0.000239523	5.0908E-06	-47.050216	1.018E-271	-0.0002495	-0.0002295	-0.0002495	-0.0002295

Based on the regression analysis, we have, for

$$MA(50)_i = \alpha \cdot hashRate_{i-50} + \beta \cdot marketCap_{i-50} + \gamma$$

$$\alpha = 0.000526513738900334$$

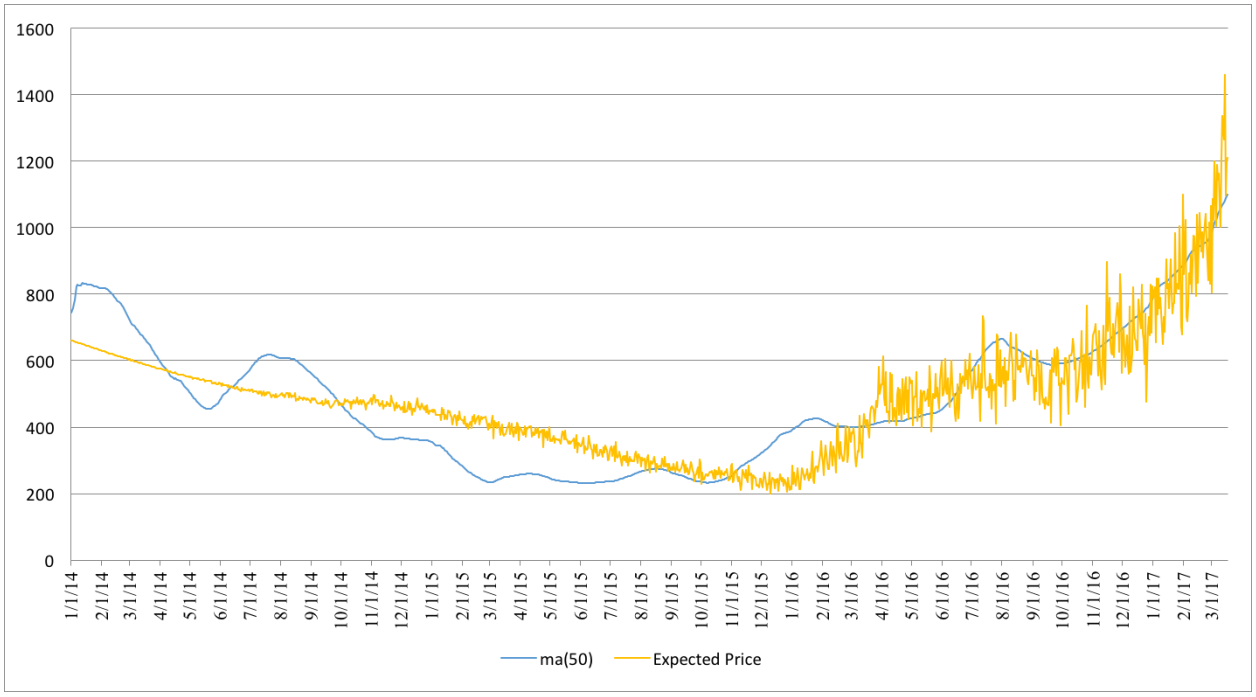
$$\beta = -0.00023952288238909$$

$$\gamma = 3530.08911436625$$

We use the equation to calculate the expected prices based on the hash rate and the market cap 50 days ago, we obtained the following csv file, with the first few rows shown below.

	date	ma(50)	hash_rate	total_cap	Expected Price
2	1/1/14	745.454163	4139.94817	11990400	660.293685
3	1/2/14	757.061522	4978.09719	11994475	659.7589262
4	1/3/14	770.89617	5847.42683	11999375	659.0429781
5	1/4/14	781.688949	4968.79793	12004200	657.42467
6	1/5/14	801.716088	4696.11999	12008300	656.2990575
7	1/6/14	824.200894	4938.50038	12012175	655.4985229
8	1/7/14	829.024166	4484.03716	12016250	654.283186
9	1/8/14	826.601623	5029.39303	12019950	653.6840887
10	1/9/14	825.344289	4908.20283	12024100	652.6262605
11	1/10/14	825.75343	5211.17832	12028150	651.8157135

Then, we plot our expected price (calculated from data 50 days before the day is is plotted) in yellow and $MA(50)$ in blue.



Since the yellow line (the our estimation based on the data 50 days ago) can somewhat predict the blue line (the real data on that day, according to our price/value definition), we believe our hypothesis works, but more data and analysis are needed to strengthen our hypothesis.

Macroeconomics Approach

We will use the macroeconomic indicators that are likely to correlate with the prices of Bitcoin. We have specifically chosen the currency exchange rates of Brazilian Real, Great Britain Pound, Euro, and Chinese Yuan in terms of one Bitcoin.

When we measure the prices of Bitcoin, we do it in terms of the US Dollar price of one Bitcoin. Then we run a linear regression to predict the US Dollar price of one Bitcoin from the prices of Brazilian Real (BRL), Great Britain Pound (GBP), Euro (EUR), and Chinese Yuan (CNY) for one Bitcoin (BTC).

Since we can transform freely from BRL-BTC to BRL-USD, GBP-BTC to GBP-USD, EUR-BTC to EUR-BTC, and CNY-BTC to CNY-USD, doing a regression in USD-BTC versus BRL-BTC, GBP-BTC, EUR-BTC, and CNY-BTC will give us the same insight as doing a regression in USD-BTC versus BRL-USD, GBP-USD, EUR-USD, and CNY-USD.

To predict the future price of USD-BTC, we first find the 7-day price average of each of the four currencies we have chosen (BRL, GBP, EUR, and CNY). Since we are trying to formulate a

model that can predict the future prices based on historical data, we will use the 7-day price average of the past 7 days excluding the day of the USD-BTC price that we want to predict.

As a result, we will have 5 columns of data to run the regression analysis, namely the USD-BTC, and the 7-day averages of BRL-BTC, GBP-BTC, EUR-BTC, and CNY-BTC of the past 7 days excluding the day of USD-BTC in the same row.

Thus we arrive at the formula:

$$E(b) = f(p1, p2, p3, p4)$$

where:

$$b = USD - BTC\ price$$
$$p1 = 7 - day\ average\ of\ BRL - BTC$$
$$p1 = 7 - day\ average\ of\ GBP - BTC$$
$$p1 = 7 - day\ average\ of\ EUR - BTC$$
$$p1 = 7 - day\ average\ of\ CNY - BTC$$

As we run a linear regression, we turn the formula into:

$$E(b) = \beta 1 p1 + \beta 2 p2 + \beta 3 p3 + \beta 4 p4$$

Where $\beta 1$, $\beta 2$, $\beta 3$, and $\beta 4$ are coefficients that we will find from the regression.

By running a linear regression, we find the following result:

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.99275942							
R Square	0.98557126							
Adjusted R Sc	0.98551772							
Standard Erro	21.6922162							
Observations	1083							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	4	34648649.9	8662162.47	18408.5032	0			
Residual	1078	507255.317	470.552242					
Total	1082	35155905.2						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1.00921895	2.49277212	0.40485808	0.68566215	-3.8820163	5.90045424	-3.8820163	5.90045424
X Variable 1	-0.0677841	0.00350643	-19.331395	1.0197E-71	-0.0746643	-0.0609039	-0.0746643	-0.0609039
X Variable 2	-0.2706486	0.04769237	-5.6748817	1.7827E-08	-0.364229	-0.1770682	-0.364229	-0.1770682
X Variable 3	0.56455821	0.05600775	10.0800009	6.7448E-23	0.45466164	0.67445478	0.45466164	0.67445478
X Variable 4	0.14534368	0.00377064	38.546197	4.964E-205	0.13794507	0.1527423	0.13794507	0.1527423

Thus, we can find that $\beta 1$, $\beta 2$, $\beta 3$, and $\beta 4$ are -0.0677841, -0.2706486, 0.56455821, and 0.14534368 respectively. We can then arrive at the USD-BTC prediction formula as:

$$E(b) = (-0.0677841)p1 + (-0.2706486)p2 + (0.56455821)p3 + (0.14534368)p4$$

We tested this formula with the historical data, to find how accurately our formula resembles the historical USD-BTC data. The result is as follows:

Therefore, we can say that our formula tracks the real historical USD-BTC price very closely.

To be more precise about how accurate our formula is, we can use R-square. The R-square from our regression analysis is 0.98557126, a number very close to 1. This means our formula is pretty accurate.

Discussions

a. What is hardest part of the project that you've encountered so far?
   I think the hardest part we've encountered so far is coming up with the functions we want for our data analysis. We have some econ knowledge but lack a bit on machine learning algorithms to come up with concrete analysis. We were able to however come up with a trading trend approach and a computational power analysis approach to visualize data. We plan on redoing them again with D3, but to show the results we use Excel.
b. What are your initial insights?
   We initially came up with a few ways to predict the future USD-BTC price from historical data. Then after some thoughts, we concluded that the Computational Power Approach

and Macroeconomics Approach would work best to track the USD-BTC price from historical data with precision. We expected the result to be a little way off at times when there were volatile fluctuations, but satisfactory in the overall trend.

c. Are there any concrete results you can show at this point? If not, why not?
   For the macroeconomics model we did some regression on bitcoin USD rate price vs Bitcoin's price rate in other currencies and found good results. The predicted price was close to the real price and can be seen in the figure.
d. Going forward, what are the current biggest problems you're facing?
   Some of the problems we will face later will be when we need to use regression and machine learning algorithms. For this midterm report, to have some analysis to show, we ended up doing two linear regression for the computational power analysis with an excel plugin.
e. Do you think you are on track with your project? If not, what parts do you need to dedicate more time to?
   The macroeconomic model seems to be giving great results and I think we need to try and understand more about this model. We need to test this model on larger datasets with different currencies and see if it still supports our hypothesis.
f. Given your initial exploration of the data, is it worth proceeding with your project, why? If not, how are you going to change your project and why do you think it's better than your current results?
   The macroeconomic model seems to be giving great results so I think we are in right track and proceed forward.