MODELING RISK IN THE CRYPTOCURRENCY UNIVERSE

ETALE INC.

Abstract. We describe a linear model that explains correlation among historical log-returns of Bitcoin-denominated cryptocurrency prices.

I. Introduction

The goal of our work is present the first attempt to understand covariance among cryptoassets. Since the legendary intuition of Markowitz, estimating correlations among any assets has been a central topic of research and interest for all those involved in financial markets, being the first and most important step in constructing a well-balanced portfolio. As it happens quite often in science, naive estimations tend to lead to surprisingly bad results, and the source of failure is in general hard to understand. In the specific case of assets returns, assume we wish to understand volatility among assets $A_1, \cdots, A_n$. For each $i = 1, \cdots, n$ we are provided with a time-series $r_{i,1}, \cdots, r_{i,\tau}$ of historical returns, sampled periodically over τ periods of time. In this case, one would consider a matrix $R = (r_{i,t})$ and attempt to naively estimate covariances by simply computing $C = R^T \cdot R$, or an adjusted variand thereof to deal with outstanding means. Upon some further reflection, interpreting C as the correct covariance matrix turns out to be a poor idea: the size of C is such that this matrix is going to be poorly conditioned. For example if $n > \tau$ the matrix itself will not be invertible, and therefore lead to portfolios that are apparently riskless. Moreover, even in the good quality data scenario $\tau = n$, correlation is not stationary as the market structure changes over time. In this type of situations, one is led to think that an intermediate step in between estimating the covariance matrix C, and blindly interpreting it, is to somehow organize a dimensionality reduction: form a hierarchical structure of clusters as dictated by the structure of C, and then interpret correlations using the hierarchy. This line of thought has many advantages, ranging from robustness to interpretability. We will implement this idea by fitting a multi-factor linear model. This means that we approximate asset returns using a linear subspace S of dimension $d << n$. The poorly conditioned

nature of C suggests that there exists $d << n$ such that virtually all the variance of asset returns can be explained via S. One simple-minded technique to achieve this goal is to use principal component analysis (PCA), which simply means to define S as the span of the d eigenvectors of C whose eigenvalues are highest. This method, albeit gaining in robustness, might still lack an adequate amount of interpretability - S looks very artificial. We finally arrive to the notion of multi-factor model. Instead of blindly picking S as the result of PCA, we add our key intuition of the real world. We know that there are some simple factors that are naturally significant sources of risk, and whose value should naturally cluster assets returns together. For example, the size or total value of an asset may be such factor; the frequency of trades may be another factor. An important technical condition that every factor should satisfy is: it should depend on quantities that change very slowly in time. For this reason one tends to pick factors that can be estimated as averages/max/min over long periods of time. Assume for each asset i and time t, we have an estimate of d factors, *i.e.* $X_{1,i,t}, \cdots, X_{d,i,t}$. A multi-factor model is then obtained by regressing retruns against factors:

$$r_{i,t} = \sum_{k=1}^{d} \beta_{t,k} X_{k,i,t} + \epsilon_{i,t} \tag{1}$$

where the intercept is taken into account by assuming $X_{d,i,t} = 1$ for every i, t. The slow-varying nature of factors in time leads to our crucial estimate:

$$R^T \cdot R \sim X^T (\beta_\tau^T \cdot \beta_\tau) X + \mathrm{diag}(\epsilon^2) \tag{2}$$

This approach has many visible advantages:

- Enhanced stability, since eigenvalues of the right-hand-side are no smaller than the entries of $\mathrm{diag}(\epsilon^2)$.
- Robustness against outliers and missing data points.
- Simple and intuitive understanding of individual factor loadings $\beta_{t,k}$.
- Flexibility of the model, in that different combinations of factors can be tested, which allows to view asset returns from different angles.

The discussion above applies to any family of assets. From now on, we focus specifically on the universe of cryptoassets. For our experiments we decided to select few very simple and intuitive factors that have been widely used in financial modeling, leaving aside more exotic ones that may be relevant to the crypto world - factors such as price of electricity,

number of miners on the blockchain, amount of activity on GitHub, sentiment analysis of Twitter data, and so on.

II. Our model

We can now dive into more technical details about our model. We consider a universe of 33 coins, that have been selected according to two criteria: high market cap; availability of historical market data. For each such coin we look exclusively at transactions to BTC: exchange rates are with respect to BTC, volumes are volumes of coins traded with BTC over all exchanges. Coins we will consider are:
'BTC' 'ETH' 'XRP' 'BCH' 'EOS' 'XLM' 'LTC' 'ADA' 'XMR' 'IOTA' 'TRX' 'ETC' 'DASH' 'NEO' 'XEM' 'BNB' 'ZEC' 'OMG' 'LSK' 'ZRX' 'QTUM' 'DOGE' 'BTS' 'DGB' 'ICX' 'STEEM' 'AE' 'WAVES' 'SC' 'REP' 'PPT' 'GNT' 'STRAT'
Let c denote any of the about coins and $T^* = [T_0, T_1]$ a time interval of T days. Let $P_c = (p_{c,1}, ..., p_{c,T})$ denote the vector of daily exchange rates for coin c with respect to BTC over period T^*. Likewise let $R_c = (r_{c,1}, ..., r_{c_T})$ denote the vector of daily returns, $S_c = (s_{c,1}, ..., s_{c,T})$ the vector of daily number of coins outstanding, and $V_c = (v_{c,1}, ..., v_{c,T})$ the vector of daily traded volumes in BTC. Our risk factors are:

- Standard deviation of returns $\text{std}(R)$.
- Strength of returns
$$\sum_{t=1}^{n} \log(1 + r_{c,t})$$
- High-low of rates
$$\log(\frac{\max_t p_{c,t}}{\min_t p_{c,t}})$$
- Average log market cap
$$\frac{\sum_{t=1}^{T} \log(p_{c,t} \times s_{c,t})}{T}$$
- Volume turnover
$$\frac{\sum_{t=1}^{T} v_{c,t}}{(\frac{\sum_{t=1}^{T} s_{c,t}}{T})}$$

Denote by $X_{k,c,t}$ the value of factor k at time t for coin c. Our model estimates returns as a linear combination:

$$r_{c,t} = \sum_{k=1}^{5} \beta_{t,k} X_{k,c,t} + \epsilon_{c,t} \tag{3}$$

Where factor loadings $\beta_{t,k}$ are obtained through weighted linear regression, and $\epsilon_{c,t}$ is an unpredictable error term. Observe that regressing using the method of ordinary least squares assumes implicitly that $\epsilon_{c,t}$ are independent and identically distributed. In particular, this method assumes that the variance of time-series $(\epsilon_{c,t})_{t\in T^*}$, estimated as $\mathrm{var}((\epsilon_{c,t})_{t\in T^*}) \sim \mathrm{var}((R_c))$, is independent of the coin c. This independence of variance is not detected here, i.e. we are in presence of heteroskedasticity. In such situation it is more appropriate to use the method of weighted least squares, which is an ordinary least squares regression using time-series $\frac{R_c}{\mathrm{std}(R_c)}$ and $\frac{(X_{k,c,t})_{t\in T^*}}{\mathrm{std}(R_c)}$. The reason for this is that dividing by $\mathrm{std}(R_c)$ normalizes the error terms to have the same variance. Summing up, our error terms and factor loadings are estimated through the following ordinary least squares problem:

$$\frac{r_{c,t}}{\mathrm{std}(R_c)} = \sum_{k=1}^{5} \beta_{t,k} \frac{X_{k,c,t}}{\mathrm{std}(R_c)} + \epsilon_{c,t} \tag{4}$$

The reader may observe that our definition of factors are slightly different from those commonly encountered in the literature. In particular,the following factors have been changed: first, we look at log market cap averaged over T^*, rather than log market cap at time T; second, we look at volume turnover, not turnover, the difference being that in our numerator we do not have number of coins traded, but the amount of coins traded expressed in BTC - in particular, the exchange rate between our coin and BTC is embedded in volume turnover. The reason for this slight difference is two-fold. First, factors computed using time-wise averages tend to vary much less over time. Second, our linear model performed better when implemented with these definitions than with classical ones.

In detail, we considered 140 time intervals, of the form $[T_0, T_1], [T_0+1\mathrm{day}, T_1+1\mathrm{day}], ..., [T_0+139\mathrm{days}, T_1+139\mathrm{days}]$ with T_0 =January 07 2018, and T_1 =April 01 2018. The model utilizing our definitions beat the model utilizing classical definitions by 89-51, where the comparison is by way of R^2 score of linear regressions. The linear model with our definition obtains, in the above-mentioned time intervals, a huge spectrum of R^2 scores, reaching a highest of 62% and averaging 22%.

The following heatmaps represent correlations between coin returns computed in the last time interval, namely May 26 2018 to August 18 2018. The first heatmap shows raw correlations, while the second shows those computed by our model.

FIGURE 1. Raw correlations

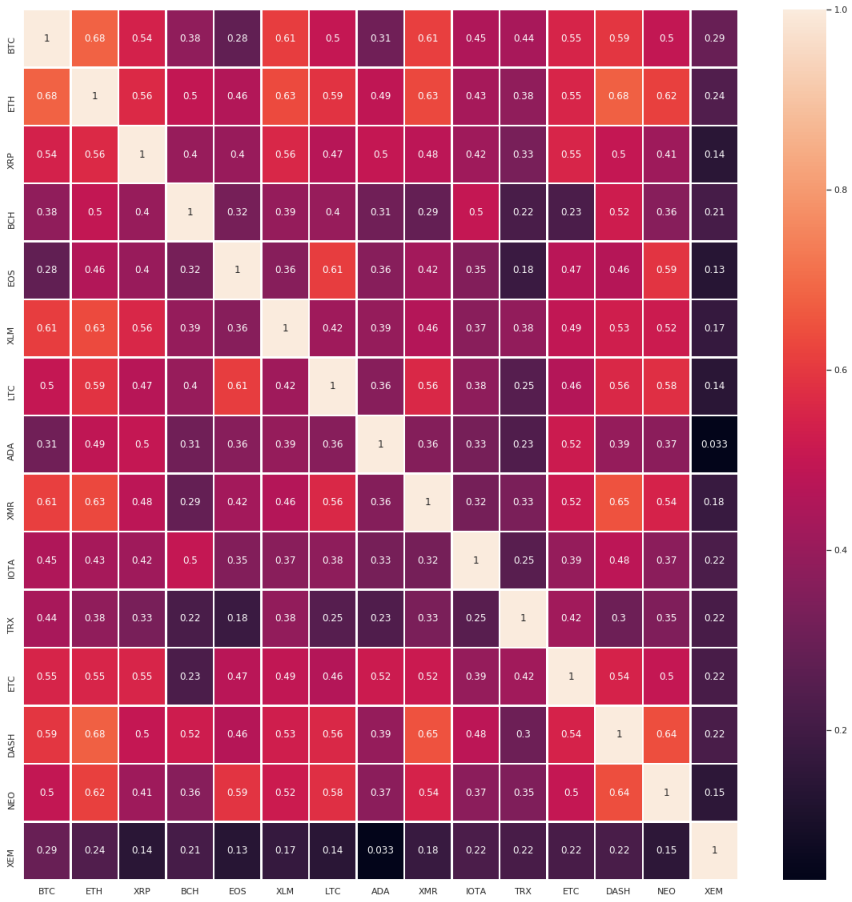
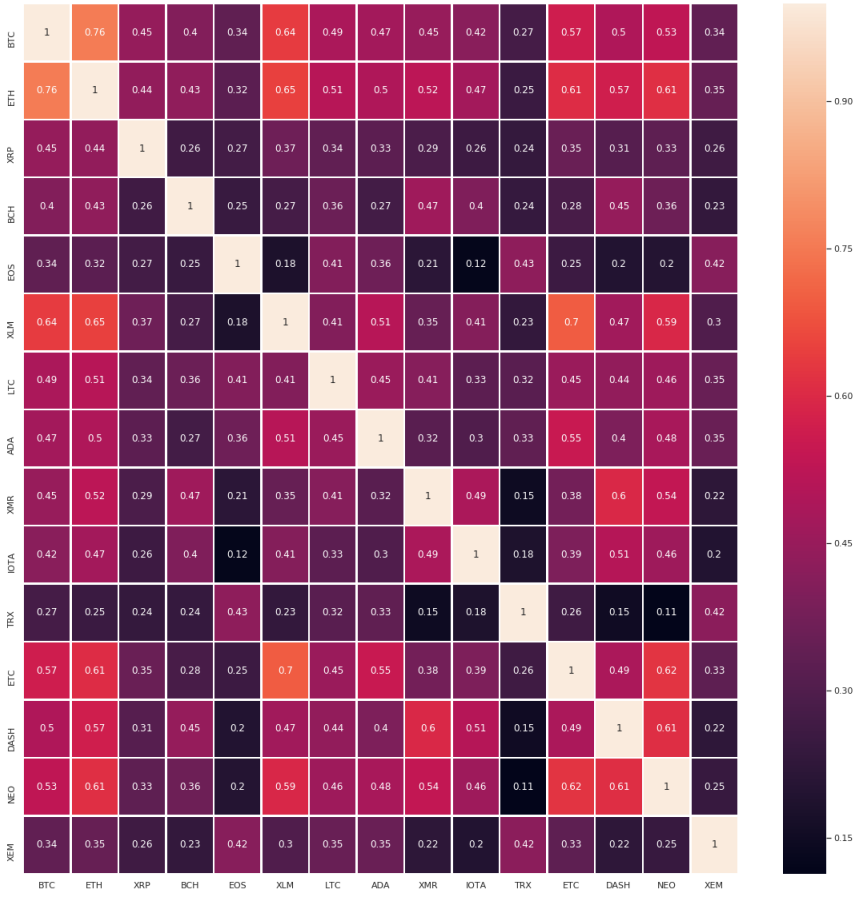


FIGURE 2. Model correlations



III. PORTFOLIO HEDGING

We can describe a first application of our method, namely construction of optimal portfolios. The methodology goes back to Markowitz. Given a time T, a collection of coins C, we assume a vector of estimated coin returns, at time T, is given and denoted $\alpha = (\alpha_c)_{c \in C}$. Moreover, let Σ denote the covariance matrix for historical coin returns up to time T. Our

goal is to invest a unit amount of capital minimizing risk and for a given value of expected return μ. We wish to find a vector of weights $w = (w_c)_{c\in C}$ solving the following quadratic optimization problem:

Minimize $w \cdot \Sigma \cdot w^T$ *under constraints* $\sum_c w_c = 1$ *and* $\sum_c \alpha_c w_c = \mu$.

This problem is easily solved via Lagrange multipliers, and it is equivalent to:

Minimize $w \cdot \Sigma \cdot w^T - \lambda\alpha \cdot w$ *under constraint* $\sum_c w_c = 1$.

The relation between λ and μ in the above formulations can be easily computed. As an example, say we expect ETH to enjoy a superior return on August 18 2018, i.e. $\alpha = (1, 0, ..., 0)$. We would like to hedge our portfolio under this expectation. Setting $\lambda = 1$ and using correlations estimated between May 26 2018 to August 18 2018, we get the following array of weights w - where a negative weight suggests shorting the coin:

[2.96419788e+00, -1.31286101e+00, -1.42622432e-01, -1.71993730e-01,-1.92916562e-01,
-5.01830577e-01, -2.82954203e-01, -1.58491848e-01,-2.25927793e-01, -6.89386864e-02,
-2.07978931e-02, -2.34667725e-01,-2.41443006e-01, -1.85703285e-01, 1.82055275e-03,
2.23332396e-02,3.95347199e-02, 5.07720836e-02, 5.51865169e-02, 5.26010066e-03,
2.98914250e-01, 1.09600168e-01, 1.06205293e-01, 8.21228399e-01, 1.07886030e-01,
7.02477857e-02, 1.30482406e-01, 1.21897933e-01,1.09311449e-01, 1.32140393e-01,
1.45112383e-01, 1.88122726e-01]

In order to test our portfolio construction, we computed optimal portfolios with $\alpha = (1, ...1, 0, ..., 0)$ - where the number of 1's in α ranges from 1 to 9 - and across the 140 days mentioned in the previous section. Remarkably, our portfolio often beats the one constructed using raw correlations, when its total return is estimated against historical data. In detail, for every such α and day T, we solve two quadratic optimization problems: in the first one Σ is estimated using our model; in the second one Σ is estimated as the correlation matrix of raw historical returns. Let us denote the resulting weight vectors by w_α and by w_α^{raw}. Finally, let R_T be the vector of realized historical returns between T and T+1day. Then return realized by our portfolio is $\mu(\alpha, T) = R_T \cdot w_\alpha$, while that realized by the raw portfolio is $\mu^{\text{raw}}(\alpha, T) = R_T \cdot w_\alpha^{\text{raw}}$. Out of these $140 \times 9 = 1260$ experiments, our porfolio beats the raw one 689-571, i.e $\mu(\alpha, T) > \mu^{\text{raw}}(\alpha, T)$ holds for 689 pairs (α, T).

This suggests that our model captures very well the correlation between top coins: 'ETH' 'XRP' 'BCH' 'EOS' 'XLM' 'LTC' 'ADA' 'XMR' 'IOTA'.

IV. Stat-arb strategy