# Portfolio Optimization Based on a Complex Networks Model

**TODO: propose a better title**

Lovre Mrčela et al.

***Abstract*— A new algorithm for portfolio optimization is presented which is based on statistical arbitrage, with potential method used to obtain the most preferred assets. A graph that represents preference relations among financial assets (i.e., if an edge exists going from asset A to asset B, then A is preferred over B) is constructed at each time step a, using the modified version of statistical arbitrage. Then, the preference flow of each asset is calculated, using the potential method[1], from which the most preferred assets are selected into the portfolio for each time step.**

**Method has been tested on dataset XY, and. . . (TODO: what happened)**

## I. INTRODUCTION

The task of portfolio optimization is to try to enhance various criteria, which most of the time include maximization of expected return and minimization of deviation. . .

The approach that is taken in this paper relies on finding abrupt deviations of price relations between in the observed set of assets. Algorithm works on pairs of assets, looking for such uncommon deviations, so generally it is expected to perform better where there is larger number of assets.

## II. KEY COMPONENTS

Following are the descriptions of two key components in the algorithm: graph of preference relations, and choosing assets for the portfolio...

### A. Graph of preference relations

This is a graph whose vertices represent assets and edges represent how much more is one asset preferred over the other. An example is shown on the Fig. 1. The measure of this preference is determined by a statistical arbitrage algorithm, and it corresponds to the magnitude of two assets' price ratio going out of what is statistically considered 'normal' range. This is illustrated on Fig. 2.

### B. Preference flow

TODO

Include measure of inconsistency?

## III. ALGORITHM

Parameters of the algorithm are:

- $T$ - length of the past time window
- $\alpha$ - the deviation threshold

Let there be total of $N$ assets in $D$ days. Let price of asset $i$ at the time step $t$ be $a_i^{(t)}$, for $i \in 1..N$ and $t \in 0..D-1$. The log prices $b_i^{(t)}$, log price differences $c_{i,j}^{(t)}$ between assets $i$ and $j$, and rolling means $m_{i,j}^{(t)}$ and standard deviations $d_{i,j}^{(t)}$ of

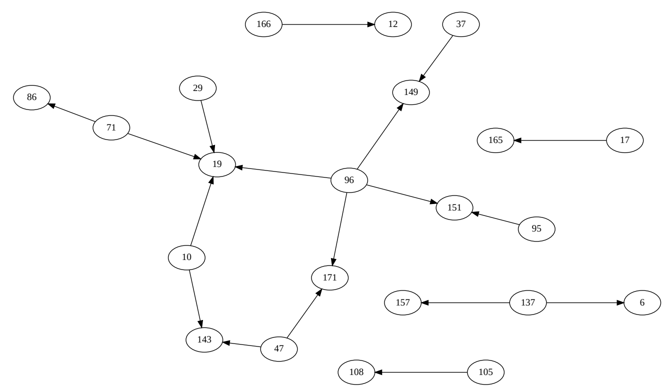


Fig. 1. TODO: put image with weights on edges

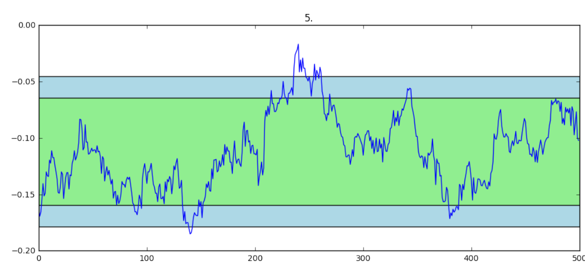


Fig. 2. TODO: put a better image, this is just for a placeholder

log price differences over past time window of size $T$ are obtained as follows:

$$b_i^{(t)} = \log\left(a_i^{(t)}\right), \tag{1}$$

$$c_{i,j}^{(t)} = b_i^{(t)} - b_j^{(t)}, \tag{2}$$

$$m_{i,j}^{(t)} = \frac{1}{T} \sum_{\tau=t-T+1}^{t} c_{i,j}^{(\tau)}, \tag{3}$$

$$d_{i,j}^{(t)} = \sqrt{\frac{1}{T} \sum_{\tau=t-T+1}^{t} \left(c_{i,j}^{(\tau)} - m_{i,j}^{(t)}\right)^2}. \tag{4}$$

Note that calculating means and standard deviations of log price differences separately for each time step $t$ is rather computationally inefficient when dealing with rolling windows of data. Therefore, it is advisable to use a rolling algorithm as described in the appendix. On that note, $c_{i,j}^{(t)}$, $m_{i,j}^{(t)}$, and $d_{i,j}^{(t)}$ may be more efficiently stored contiguously in memory as a matrix, using following coding scheme: a pair $(i, j)$, where $i < j$, should be encoded to $k$ as:

$$k = N \cdot (i-1) + j - 1 - i \cdot (i+1)/2\,, \tag{5}$$

and decoded from $k$ as:

$$i = \left\lfloor N + 1/2 - \sqrt{(N+1/2)^2 - 2(N+k)} \right\rfloor, \tag{6}$$

$$j = k + i \cdot (i+1)/2 - N \cdot (i-1) + 1. \tag{7}$$

An example of proposed coding is shown on figure 3.

| $i/j$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | · | 0 | 1 | 2 | 3 |
| 2 | · | · | 4 | 5 | 6 |
| 3 | · | · | · | 7 | 8 |
| 4 | · | · | · | · | 9 |
| 5 | · | · | · | · | · |

| $k$ | $i$ | $j$ |
|---|---|---|
| 0 | 1 | 2 |
| 1 | 1 | 3 |
| 2 | 1 | 4 |
| 3 | 1 | 5 |
| 4 | 2 | 3 |
| 5 | 2 | 4 |
| 6 | 2 | 5 |
| 7 | 3 | 4 |
| 8 | 3 | 5 |
| 9 | 4 | 5 |

Fig. 3. Example of the proposed coding scheme, for $N = 5$. A dot $(\cdot)$ indicates that that combination is not used.

### A. Creating the graph

Using the obtained $c_{i,j}^{(t)}$, $m_{i,j}^{(t)}$, and $d_{i,j}^{(t)}$ it is now possible to create a graph of preference relations between assets for each time step $t$. Considering the time step $t$, we find all such pairs of assets $(i, j)$ for which holds:

$$\left| c_{i,j}^{(t)} - m_{i,j}^{(t-1)} \right| > \alpha \cdot d_{i,j}^{(t-1)}, \tag{8}$$

i.e. current log price difference is at least $\alpha$ deviations distant from mean value of the past time window.

Afterwards, for each observed pair $(i, j)$ that breaks the threshold we add into graph vertices $i$ and $j$, and a weighed edge going from $i$ to $j$, with weight $w_{i,j}^{(t)}$ obtained as:

$$w_{i,j}^{(t)} = \left( c_{i,j}^{(t)} - m_{i,j}^{(t-1)} \right) \Big/ d_{i,j}^{(t-1)}. \tag{9}$$

Thus, it is possible to create a relatively sparse graph for each time step $t \in T..D-1$. At some time steps it is possible that the graph could be empty, if it is the case that no pair $(i, j)$ satisfies (8). Lower values of parameter $\alpha$ yield denser graphs.

### B. Choosing assets from graph

From previously obtained graph it is possible to tell which pair of assets has the highest preference flow. However it is not yet possible to directly tell which is the most preferrable asset, or obtain the measure of preference for individual assets. For this we need to calculate some sort of potential for each node in graph, which corresponds to difference in amount of flow going in and out of the node. A quick way to do this is by using the potential method as described in [1]. The summary of the method is as follows:

1) Let $\mathbf{A}$ be the incidence matrix of previously obtained graph. $\mathbf{A}$ has following properties:
   a) each row corresponds to an edge in the graph, and each column to a node,
   b) for every edge in the graph going from node $i$ to node $j$, there is a corresponding row that has $-1$ and 1 in columns that correspond to nodes $i$ and $j$, respectively,
   c) the remainder of elements in the matrix are zeros.
2) Let $\mathbf{F}$ be column vector that contains edge weights, and order of the edges is consistent with order of the edges in $\mathbf{A}$.
3) Let $\mathbf{X}$ be column vector that contains potentials of each node, and order of the nodes is consistent with order of the nodes in $\mathbf{A}$.
4) Now $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{F}$ should satisfy the equation

$$\mathbf{AX} = \mathbf{F}, \tag{10}$$

meaning that the difference between potential of any two nodes should result result in weight of the edge between them. Most of the time this represents an overdetermined system since $\mathbf{A}$ has more rows than columns, so we convert it to a least squares problem:

$$\min_{\mathbf{X}} \left\{ \|\mathbf{AX} - \mathbf{F}\|^2 \right\} \Rightarrow \frac{\partial}{\partial \mathbf{X}} \|\mathbf{AX} - \mathbf{F}\|^2 = \mathbf{0}, \tag{11}$$

from which we obtain the equation:

$$\mathbf{A}^T\mathbf{AX} = \mathbf{A}^T\mathbf{F}. \tag{12}$$

Following constraint is also added:

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}}_{N} \cdot \mathbf{X} = 0 \tag{13}$$

to ensure that total amounts of positive and negative potential are equal.

5) Joining the previous two equations together by adding (13) to each row of (11) results in:

$$\begin{gathered} \mathbf{A}^T\mathbf{AX} + \mathbf{JX} = \mathbf{A}^T\mathbf{F} \\ \left[\mathbf{A}^T\mathbf{A} + \mathbf{J}\right]\mathbf{X} = \mathbf{A}^T\mathbf{F}, \end{gathered} \tag{14}$$

where $\mathbf{J}$ is a matrix of ones with same dimension as $\mathbf{A}^T\mathbf{A}$. Finally, solving for $\mathbf{X}$ gives us:

$$\mathbf{X} = \left[\mathbf{A}^T\mathbf{A} + \mathbf{J}\right]^{-1}\mathbf{A}^T\mathbf{F}. \tag{15}$$

## IV. RESULTS

TODO

## V. CONCLUSIONS

TODO

## APPENDIX

### A. Rolling mean and variance algorithm

TODO

## REFERENCES

[1] L. Čaklović, Decision Making by Potential Method