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 flow 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 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 (TODO: which dataset), by simulating the portfolio obtained by the algorithm, including the trading cost of 0.1%. Sharpe ratios over 1.0 were recorded.

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.

II. CONCEPTS AND METHODS

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

A. Preference relation

Preference relation is a *strict weak ordering* that corresponds to the way humans prefer some entity over another. This relation is specific in that it is:

- *irreflexive* — every entity is not preferrable over itself,
- *asymmetrical* — if x is preferrable over some y, then y is not preferrable over x,
- *transitive* — if x is preferrable over y, and y is preferrable over z, then x is also preferrable over z,
- *transitive in incomparability* — noting that x and y may be incomparable (i.e., neither x is preferrable over y, nor y is preferrable over x): if x is incomparable with y, and y is incomparable with z, then x is also incomparable with z.

We naturally impose this kind of relation when describing relationships among the assets.

B. Graph of preference flow

This is a graph whose nodes represent assets, and edges (directed) describe how much more is one asset preferred over the other. In case of missing edge between two assets, we consider that neither is preferable over another. The graph as a whole describes preference flow among its nodes. Relations in this graph should ideally be in compliance with properties of previously mentioned properties of preference relation. However, they might also be inconsistent — this is dealt with in the later stage of algorithm. An example of graph is shown on Fig. 1.

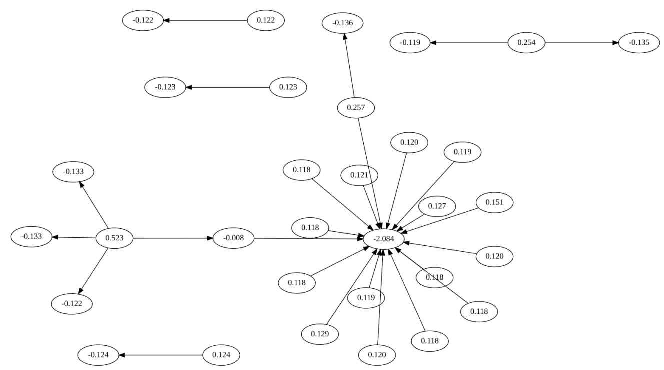

Fig. 1. (TODO: put image with weights on edges) An example of a graph of preference flow. Asset number and calculated preference are inscribed in each node, and preference flows are shown on edges.

The measure of this preference flow is determined by a statistical arbitrage algorithm, and it corresponds to the magnitude of a pair of assets prices ratio going out of what is considered statistically confident range. This is illustrated on Fig. 2.

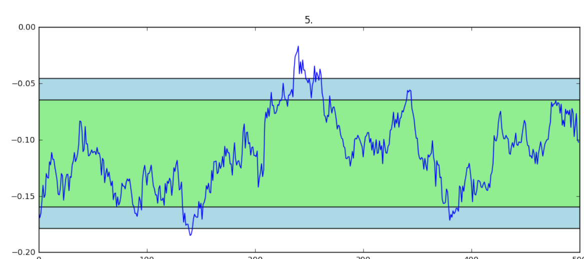

Fig. 2. (TODO: put a better image, this is just for a placeholder) Log price difference of a pair of assets during period of $T+1$ time steps. During the last observed timestep, it goes (TODO: broj) times standard deviations away from mean value of past time window of size T. This would be measure of preference flow from asset with higher price to the asset with lower price.

The main idea of preference flow is that preference of a particular node depends on amount of preference flowing in and out of it. This allows for some kind of generalized statistical arbitrage over multiple assets at a time.

C. Potential method

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 are the most or least preferrable assets, or obtain the measure of preference for individual assets. For calculating the preference of some node in the graph, we use the potential method[1]. The potential of a node corresponds to difference in amount of flow going in and out of the node.

A concise summary of the method is as follows:

1) For the observed graph, let there be a total of N nodes and E edges.
2) Let $\mathbf{A}$ be the incidence matrix $[E \times N]$ 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.
3) Let $\mathbf{F}$ be column vector $[E \times 1]$ that contains edge weights, and order of the edges is consistent with order of the edges in $\mathbf{A}$.
4) Let $\mathbf{X}$ be column vector $[N \times 1]$ that contains potentials of each node, and order of the nodes is consistent with order of the nodes in $\mathbf{A}$.
5) Now $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{F}$ should satisfy the equation
$$\mathbf{AX} = \mathbf{F}, \tag{1}$$
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:
$$\min_{\mathbf{X}} \left\{ \|\mathbf{AX} - \mathbf{F}\|^2 \right\} \Rightarrow \frac{\partial}{\partial \mathbf{X}} \|\mathbf{AX} - \mathbf{F}\|^2 = \mathbf{0}, \tag{2}$$
from which we obtain the equation:
$$\mathbf{A}^T\mathbf{AX} = \mathbf{A}^T\mathbf{F}. \tag{3}$$
Following constraint is also included:
$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}}_{N} \cdot \mathbf{X} = 0 \tag{4}$$
to ensure an unique solution and that total amounts of positive and negative potential are equal.
6) Joining the previous two equations together by adding (4) to each row of (3) results in:
$$\begin{aligned} \mathbf{A}^T\mathbf{AX} + \mathbf{JX} &= \mathbf{A}^T\mathbf{F} \\ \left[\mathbf{A}^T\mathbf{A} + \mathbf{J}\right]\mathbf{X} &= \mathbf{A}^T\mathbf{F}, \end{aligned} \tag{5}$$
where $\mathbf{J}$ is a matrix of ones with same dimension as $\mathbf{A}^T\mathbf{A}$. Finally, solving for $\mathbf{X}$ gives us:
$$\mathbf{X} = \left[\mathbf{A}^T\mathbf{A} + \mathbf{J}\right]^{-1} \mathbf{A}^T\mathbf{F}. \tag{6}$$
7) Furthermore, it has been shown that in case of a complete graph term $\left[\mathbf{A}^T\mathbf{A} + \mathbf{J}\right]^{-1}$ will be equal to $\frac{1}{N}\mathbf{I}$ due to its connection to the Laplace matrix. In our case, graphs will be incomplete most of the time, but they may be represented as complete graphs if we treat missing edges as edges of weight 0 (direction doesn't matter). On the other hand, $\mathbf{A}^T\mathbf{F}$ doesn't change if we remove corresponding entries for missing edges. Thus, we can shorten (6) for some more:
$$\mathbf{X} = \frac{1}{N}\mathbf{A}^T\mathbf{F}, \tag{7}$$
to get as computationally optimal expression as possible.

(TODO: čini se nepotrebno komplicirano uvoditi metodu od koje na kraju samo dobijemo zbroj svih bridova koji utjeu u vor minus oni koji izlaze iz njega.)

III. ALGORITHM

Parameters of the algorithm are:

- T - length of the past time window,
- α - the deviation threshold,
- β - asset selection 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 log price differences over past time window of size T are obtained as follows:

$$b_i^{(t)} = \log\left(a_i^{(t)}\right), \tag{8}$$

$$c_{i,j}^{(t)} = b_i^{(t)} - b_j^{(t)}, \tag{9}$$

$$m_{i,j}^{(t)} = \frac{1}{T}\sum_{\tau=1}^{T} c_{i,j}^{(t-\tau)}, \tag{10}$$

$$d_{i,j}^{(t)} = \sqrt{\frac{1}{T}\sum_{\tau=1}^{T}\left(c_{i,j}^{(t-\tau)} - m_{i,j}^{(t)}\right)^2}. \tag{11}$$

Note that period over which means and standard deviations of log price differences are calculated does not include time step t. Also note that calculating them 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 if stored contiguously in memory as a matrix, using following coding scheme: a pair (i,j), where $i < j$, should be encoded to k as:

$$k = N \cdot (i-1) + j - 1 - i \cdot (i+1)/2\,, \tag{12}$$

and decoded from k as:

$$i = \left\lfloor N + 1/2 - \sqrt{(N+1/2)^2 - 2(N+k)} \right\rfloor, \tag{13}$$

$$j = k + i \cdot (i+1)/2 - N \cdot (i-1) + 1. \tag{14}$$

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 of preference flow

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 flow among assets for each time step t. Considering the time step t, we find all such pairs of assets (i, j) for which holds:

$$\left| c_{i,j}^{(t)} - m_{i,j}^{(t)} \right| > \alpha \cdot d_{i,j}^{(t)}, \tag{15}$$

i.e. current log price difference is at least α deviations distant from mean value of the past time window. Parameter α determines how many pairs of assets should constitute the graph at current time step.

Afterwards, for each observed pair (i, j) that exceeds 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:

$$w_{i,j}^{(t)} = \left(c_{i,j}^{(t)} - m_{i,j}^{(t)} \right) \Big/ d_{i,j}^{(t)} . \tag{16}$$

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 (15). Lower values of parameter α yield denser graphs.

B. Choosing assets from graph by potential method

We obtain preference of each asset via the potential method, as described earlier in II-C. Let φ_i denote the potential of asset i at the current time step. By obtaining preferences of each asset it is now possible to pick assets into the portfolio. The most preferred assets are being bought while the least preferred should be short-sold if possible.

For determining the assets that should be held in portfolio at time step t, we find such assets i which satisfy:

$$\varphi_i \geq \beta \cdot \Phi, \tag{17}$$

where Φ is $\max_j \left\{ \left| \varphi_j \right| \right\}$, and β is the asset selection threshold parameter mentioned earlier. Likewise, for short-selling we choose those assets i which satisfy:

$$\varphi_i \leq -\beta \cdot \Phi. \tag{18}$$

IV. RESULTS

Results were obtained by testing on dataset (TODO: dataset). Short selling was disabled.

T=90

α	β	Sharpe ratio	turnover rate	profit
3.0	1.0	1.12033	**0.64613**	18.63180
	0.9	1.14333	0.68407	19.03378
	0.8	**1.15759**	0.71548	**19.35692**
	0.7	1.13153	0.75715	19.00914
	0.6	1.06598	0.79802	17.70338
	0.5	1.03619	0.85536	17.16405
3.25	1.0	1.04127	0.68351	17.27340
	0.9	1.07932	0.71015	17.90707
	0.8	1.12707	0.75099	18.72864
	0.7	1.10705	0.79748	18.42081
	0.6	1.05853	0.84321	17.51641
	0.5	1.01289	0.89148	16.76236
3.5	1.0	0.99809	0.72792	16.40651
	0.9	1.02397	0.76663	16.88082
	0.8	1.03420	0.81126	17.09970
	0.7	1.05821	0.86305	17.48784
	0.6	1.01590	0.91392	16.89037
	0.5	0.98479	0.96461	16.27100
3.75	1.0	0.98829	0.80549	15.84890
	0.9	1.02361	0.84603	16.40503
	0.8	1.04266	0.89192	16.74853
	0.7	1.03920	0.95609	16.73169
	0.6	1.05219	1.00587	17.00883
	0.5	1.03340	1.06956	16.67687
4.0	1.0	1.00574	0.91970	15.02475
	0.9	1.02546	0.95710	15.37962
	0.8	1.04724	0.99747	15.74329
	0.7	1.04383	1.03972	15.76506
	0.6	1.01710	1.08639	15.39334
	0.5	1.01925	1.15608	15.43147

Fig. 4. Results. TODO: comment.

V. CONCLUSIONS

Algorithm works on pairs of assets, looking for those deviations which are uncommon, so generally it is expected to perform better where there is larger number of assets as more deviations will be discovered.

APPENDIX

A. Rolling mean and variance algorithm

TODO

REFERENCES

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

T=120				
α	β	Sharpe ratio	turnover rate	profit
3.0	1.0	1.02488	**0.57236**	17.63623
	0.9	0.99552	0.60648	17.16067
	0.8	1.00498	0.64939	17.24051
	0.7	1.01485	0.68910	17.43257
	0.6	1.04164	0.72308	17.91063
	0.5	**1.08388**	0.78830	**18.33920**
3.25	1.0	0.95261	0.60842	15.76037
	0.9	0.97329	0.63420	16.11968
	0.8	0.98852	0.67298	16.35240
	0.7	0.97404	0.72470	16.12448
	0.6	0.94128	0.77388	15.67659
	0.5	0.97199	0.81790	15.97750
3.5	1.0	0.89277	0.66365	14.63103
	0.9	0.93685	0.69743	15.37556
	0.8	0.94367	0.73989	15.52817
	0.7	0.95012	0.79145	15.73655
	0.6	0.96529	0.83751	15.97783
	0.5	0.99559	0.88986	16.08797
3.75	1.0	0.90895	0.74222	14.27841
	0.9	0.96107	0.77696	15.10384
	0.8	0.98794	0.82238	15.53080
	0.7	1.00128	0.87246	15.92569
	0.6	0.98468	0.92541	15.67132
	0.5	1.01009	0.98699	16.09083
4.0	1.0	0.99691	0.83501	14.67240
	0.9	1.01115	0.87406	14.89908
	0.8	1.06557	0.91977	15.71279
	0.7	1.06097	0.95934	15.76776
	0.6	1.03053	0.99846	15.34007
	0.5	1.04213	1.05134	15.47247

Fig. 5. Results. TODO: comment.