Aditya Pidaparti
Paper 2
12/9/16

Introduction

Modern portfolio selection theory is a branch of machine learning involving sequential decision making. Portfolio selection algorithms seek to maximize returns of financial portfolios, and do so by making decisions based on past performance of different options in the market. Here, we focus on algorithms analyzing data from the only stock market, ignoring derivatives, shorts, bonds, and other securities, and we focus on Meta Algorithms[1] and how they compare to existing machine selected portfolio methods in literature such as Exponentiated Gradient (EG)[2] and Anticor.[4]

Background

If a given portfolio algorithm can select between *N* stocks, where each stock is represented by s_i for $1 \leq i \leq N$, it can do so over *T* day for a given dataset. We assemble our dataset by letting $r_t(i)$ be the price relative in day *t* of the stock s_i, and let $\mathbf{r}_t$ then be the price relative vector (contains all the price relatives) for day *t*. A price relative greater than one indicates a gain, and a price relative less than one indicates loss. Assembling the price relative vectors across *t* days, we define $\mathbf{r}_{1:t}$ to be the matrix of price relative vectors going from day 1 to day *t*.

For each day *t*, a portfolio selection algorithm will calculate a proportion of total wealth to allocate to each stock. Much like the price relative vector $\mathbf{r}_t$, a probability distribution $\mathbf{x}_t$ is created each day, which can also be looked at as a portfolio with a wealth of $1. In parallel, to price relatives, we also define a matrix of weight vectors for all days from 1 to *t* as $\mathbf{x}_{1:t}$.

Algorithms have one goal: to maximize return. As such, we denote that return as *S*. Given a dataset with *T* days, a stock will seek to maximize the following overall return:

$$\text{Equation 1: } S(T) = \prod_{t=1}^{T}(r_t^T x_t)$$

This equation is an Online Convex Optimization (OCO)[1], where the loss can be modeled by $\varphi_t(x_t) = -\log(r_t^T x_t)$. Over *T* days, the loss is shown as

$$\text{Equation 2: } \sum_{t=1}^{T}\varphi_t(x_t) = \sum_{t=1}^{T} -\log(r_t^T x_t) = -\log(S(T))$$

Method

Aditya Pidaparti
Paper 2
12/9/16

Meta algorithms (MAs) are a new development in portfolio selection theory. They work by minimizing a convex loss function to take advantage of several base algorithms at the same time. In essence, several algorithms already in literature are taken and their portfolio selections treated as 'recommendations' for the meta algorithm. The MAs then allocate wealth such that the loss or regret function is minimized (how an algorithm determines the quality of a particular selection). Two algorithms described are Online Gradient Updates (OGU) and Online Newton Updates (ONU), which both are guaranteed to perform at least as well as the best base algorithm with a convex cost function.

The OGU meta algorithm optimizes its portfolio selection with *k* algorithms and a dataset with *T* days as follows: Weights are initialized to an equal distribution across all methods, so each algorithm is allocated $\frac{1}{k}$ of the total initial wealth. Then, for each day *t*, several steps are taken: First, each base algorithm is run and the weights of each is stored in the matrix $\mathbf{X_t}$. , we find a weight vector $\mathbf{w_t}$ such that $x_t^{w_t} = X_t w_t$. We then use the loss function φ_t as defined above and the portfolio is assigned new weights as shown:

$$\text{Equation 3: } w_{t+1,h} := \frac{w_{t,h} * \exp(-\eta l_t(h))}{Z_t}$$

Eq. 3 assumes a given learning rate η, a partition function for $\mathbf{w_t}$ given as Z_t, loss function l_t, given by:

$$\text{Equation 4: } l_t = -\frac{1}{2\bar{u}} \frac{X_t^T r_t}{r_t^T X_t w_t} + e$$

Eq. 4 defines *e* as equal to a k-element vector of ones, and $\bar{u}$ as the following for day *t*:

$$\text{Equation 5: } \bar{u} = \frac{r_{max}}{r_{min}} = \frac{maximum\ return}{minimum\ retur}$$

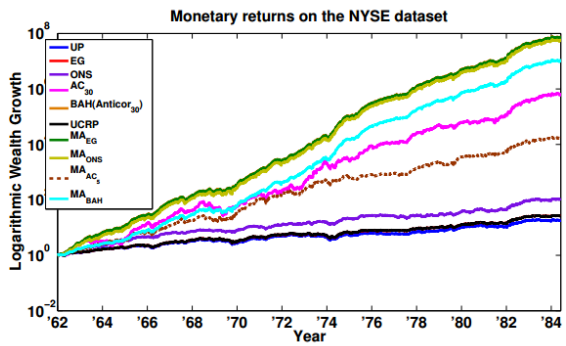
Now, both the portfolio gain from the previous day and the new portfolios have been selected.

ONU follows a similar setup as OGU, but is a different algorithm. The weights are established in the same fashion, however as opposed to a gradient setup, instead Newton steps are taken to improve the portfolio selection. A value β is defined to be $\frac{1}{8\bar{u}}$. Then for each day *t* in a dataset with *T* days, Newton steps are implemented. First, as with OGU, the iterates are computed and stored in $\boldsymbol{X_t}$ and the multiplicative gain is calculated. Next the convex function φ_t is obtained as in OGU. Finally, the distribution of weights is updated, but in a different manner. The algorithm A_t defined for day *t* as $\sum_{\tau=1}^{t} \nabla \varphi_t (X_t w_t) \nabla \varphi_t (X_t w_t)^T + \frac{k}{\beta^2 c^2} I$. The distribution is then updated using A_t:

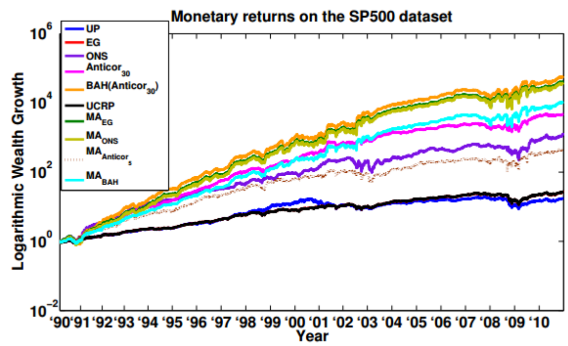
Equation 5: $w_{t+1,h} := \prod_{\Delta_k}^{A_t}\left(w_t - \frac{2}{\beta}A_t^{-1}\frac{X_t^T r_t}{r_t^T X_t w_t}\right)$

Data/Analysis

ONU and OGU are guaranteed to perform at least as well as one of their base algorithms, given that they have a convex cost function, i.e. there is a minimum regret or loss that can be found. Data shown in Figures 1(a) and 1(b) support this statement, as both the OGU and ONU methods perform better than the best individual convex algorithm.



(a) Monetary returns on NYSE.



(b) Monetary returns on S&P500.

Figure 1(a) and 1(b). As seen, MA_{EG} (OGU) and MA_{ONS} (ONU) both perform well relative to the other algorithms tested. However, they do not always perform the best.

Four meta algorithms were tested, MA_{EG}, MA_{ONS}, $MA_{Anticor}$, and MA_{BAH}. The first two reflect the algorithms discussed above, respectively OGU and ONU. The last two utilize a strategy known as anti-correlation, or Anticor.[4] Unlike MAs or the other base algorithms involved such as EG, Anticor is a 'follow-the-loser' algorithm. Anticor chooses to reallocate wealth towards stocks anti-correlated with those that perform well or above average in the market, so it chooses stocks performing poorly. However, it only analyzes data in a window *w* before the certain day, where *w* specifies the number of days into the past that the frame extends. $MA_{Anticor}$ invests in Anticor algorithms, using data going back *w* days, ranging from $2 \leq w \leq W$, for *W* being some maximum timeframe that is specified when the algorithm is run. For example, if *W* were initialized to 10, then the meta algorithm would have access to 10 Anticor algorithms, each 'expert' running and making decisions as if it had access to 10, 9, 8, … days in the past all the way down to just analyzing just yesterday and today (*w* of 2 days). Finally, MA_{BAH}, or the meta algorithm for buy and hold, takes the same setup as $MA_{Anticor}$, but after portfolio initialization, does not reallocate wealth between different 'experts'.

In Figure 1 we see the results for the various MAs and comparisons to their base algorithms. For the test, the window sizes of the Anticor algorithms was *W* = 30. For both the S&P 500 and NYSE datasets, MA_{EG} MA_{ONS}, and MA_{BAH} perform the best of all algorithms analyzed, including the base algorithms used to create the MAs, while $MA_{Anticor}$ performed better than any universal algorithm, but not better than every base one.

Discussion

Meta algorithms as described here are guaranteed to perform better than any of their constituent base algorithms, but have many constraints to achieve that performance. These include where the algorithm breaks, the algorithm assumptions, and the algorithm efficiency.

The first constraint is for any part of the MA to work, every base algorithm must always be able to create a suitable portfolio. This leaves MAs in a difficult situation if a chosen stock fails, and accounting for the scenario will depend on the implementation. In addition, the MA is limited in its guarantee to only perform as well as the best algorithm that is universal, or the best constantly rebalanced portfolio strategy in hindsight. As universal portfolios are also guaranteed to perform at least as well as the best stock, if at least one base algorithm is universal, the MA is guaranteed to perform at least as well as the best stock in its portfolio. However, as $MA_{Anticor}$ showed in results, an MA may not perform as well as its best base algorithm. $MA_{Anticor}$'s lack of success makes sense, as it implements a 'follow-the-loser' strategy, allocating resources to poor-performing 'experts', going directly against the strategy of MAs, to pick winners.

Second, Meta Algorithms rely on a few crucial assumptions. They require a convex cost function, which is the case in gradient and newton-step methods, but may not be true everywhere. In addition, the MAs have only been tested on two datasets, and haven't been tried out for true effectiveness. As no comparison was made on current markets, the MAs aren't useful for investing unless first examined when run against modern markets.

Finally, the meta algorithms suffer from potential inability to run. All the base algorithms must go through updates every day, and the complexity can make computational costs prohibitive. Only efficient portfolio update methods can be used, as slow methods that iterate every day but give excellent performance mean that fewer base algorithms in total can be used. This implies an realistic upper bound on the number of algorithms that can form meta algorithm, limited by computing power and implementation efficiency.

Conclusion

Meta algorithms are new development in portfolio selection theory. They are composed of several base algorithms, and then makes decisions based on the performance of each of these base methods. MAs are guaranteed to at least have returns of a universal algorithm, provided it is one of the base algorithms. When tested on the S&P 500 and NYSE datasets given, MAs did well and followed the best performers in the market, except for $MA_{Anticor}$. However, all outperformed the best universal algorithm in their pool, showing that meta algorithms are a new method which has been found to be more successful and a potential path to follow in portfolio selection research.

References

1. Das, P., & Banerjee, A. (2011, August). Meta optimization and its application to portfolio selection. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1163-1171). ACM
2. Helmbold, D. P., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1998). On-Line Portfolio Selection Using Multiplicative Updates. Mathematical Finance, 8(4), 325-347.
3. Das, P., Johnson, N., & Banerjee, A. (2013, June). Online Lazy Updates for Portfolio Selection with Transaction Costs. In AAAI.
4. Borodin, A., El-Yaniv, R., & Gogan, V. (2004). Can we learn to beat the best stock. Journal of Artificial Intelligence Research, 21, 579-594.