Stanford EE 378A Final Report

Nico Chaves, Dominic Delgado
June 4, 2016

I. Introduction

Markowitz and Cover offer competing philosophies toward solving the portfolio selection problem. Instead of trying to determine which approach is most desirable overall, we recognize that different approaches to portfolio selection perform better or worse depending on the market conditions, which can change dramatically over time. Based on this observation, we introduce an algorithm called "Expert Pooling using Exponential Weighting" (EPEx), which combines the portfolios of individual experts based on their recent performance.

II. Background on Individual Experts

Before discussing the EPEx algorithm in detail, we first introduce some of the underlying experts that may be aggregated together to form an EPEx portfolio.

A. Uniform Constant Rebalancing Portfolio

The Uniform Constant Rebalancing Portfolio (UCRP) invests an equal share of its current money into each stock every day. UCRP is simple and widely known, so we do not discuss any additional details here.

B. Nonparametric Markowitz

The Nonparametric Markowitz[1] (NPM) algorithm embodies the basic goal of Markowitz portfolio optimization: maximizing the expected return while keeping the variance of the return small. NPM groups stocks according to their similarity. For stock i, NPM forms the similarity set Γ_i containing the k stocks most similar to stock i in terms of the distance $d(y_i, y_j)$. Here y_i contains information about stock i over the last w days, such as its opening, low, high, and closing prices. We used the L2 norm as our distance metric. We then solve an optimization problem analogous to the Markowitz problem:

$$\max_{\boldsymbol{b}} \sum_{i=1}^{n} \left(\boldsymbol{b}^{(i)}\right)^T \boldsymbol{\mu}^{(i)} - \lambda \left(\boldsymbol{b}^{(i)}\right)^T \Sigma^{(i)} \boldsymbol{b}^{(i)} \text{ s.t. } \|\boldsymbol{b}\|_1 = 1$$

where n is the number of stocks and $\boldsymbol{b}^{(i)} \in \mathbb{R}^k$ is a vector whose elements are all given by b_i, which is the current allocation to stock i. The vector $\boldsymbol{\mu}^{(i)}$ and matrix $\Sigma^{(i)}$ are given by $G^{(i)}\boldsymbol{\mu}$ and $G^{(i)}\Sigma\left(G^{(i)}\right)^T$, respectively. Here $G^{(i)} \in \mathbb{R}^{k\times n}$, where $G^{(i)}_{qr} = 1$ if stock r is the q^{th} stock in Γ_i and $G^{(i)}_{qr} = 0$ otherwise.

We estimate $\boldsymbol{\mu}$ and Σ from the data. Our estimate of the vector $\boldsymbol{\mu}$ each day is obtained from the mean closing price of each stock. Estimating a large covariance matrix from a small number of data points can produce singular matrices, so we split the estimation of Σ into two phases. In the first phase, we estimate Σ assuming that the price relatives of the stocks are conditionally independent, which guarantees that Σ will be nonsingular (since it is diagonal). Then, after reaching day $t > t_0$ (where t_0 can be specified as a parameter), we drop the conditional independence assumption and begin updating Σ using terms of the form $\boldsymbol{x}^{(t)}(\boldsymbol{x}^{(t)})^T$, where $\boldsymbol{x}^{(t)}$ is the vector of price relatives on day t. Empirically, the warm-start with diagonal covariance does not produce singular covariance matrices.

C. Online Moving Average Reversion

Online Moving Average Reversion (OLMAR) employs an allocation strategy based on the assumption that the price of each stock on the next day will revert to the stock's mean price over a window of length w. The corresponding estimate of the next day's price relative vector is therefore given by:

$$\tilde{\boldsymbol{x}}_{t+1}(w) = \frac{1}{w}\left(1 + \frac{1}{\boldsymbol{x}_t} + ... + \frac{1}{\prod_{i=0}^{w-2} \boldsymbol{x}_{t-i}}\right)$$

where w is the window size parameter. OLMAR then updates the portfolio using a "passive aggressive" investment strategy. The algorithm's authors frame the update as the following convex optimization problem:

$$\boldsymbol{b}_{t+1} = \arg\min_{\boldsymbol{b}\in\Delta_m} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{b}_t\|^2 \text{ s.t. } \boldsymbol{b}\cdot\tilde{\boldsymbol{x}}_{t+1} \geq \epsilon$$

where Δ_m denotes the set of valid portfolios (i.e. $\Delta_m = \{\boldsymbol{b} : \|\boldsymbol{b}\|_1 = 1\}$) and ϵ is a parameter that quantifies how passive or aggressive the algorithm should be. In particular, if the predicted price relatives $\tilde{\boldsymbol{x}}_{t+1}$ suggest

[1] Although this algorithm is inspired by the discussion of nonparametric methods in [1], it is not based on any of the papers referenced in that section.

that the current portfolio will increase by a factor greater than or equal to ϵ, then $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t$ (i.e. the portfolio will remain the same). Otherwise, the portfolio will be updated by solving the optimization. Since this is a well-studied problem and its solution is provided elsewhere, we do not reproduce it here.

D. Robust Median Reversion

Robust Median Reversion (RMR) follows a very similar procedure as OLMAR, except that it uses the L1 median prices over the window to estimate the next price relatives. As a result, we do not explain the algorithm in detail here.

III. Expert Pooling using Exponential Weighting

A. Motivation

Different allocation strategies perform better or worse under different market conditions. As a result, we aggregate, or pool together, the portfolios of several independent experts based on the recent performance of each expert. We define the performance of an expert to be its Sharpe ratio over a recent period of time. We refer to this portfolio selection strategy as “Expert Pooling using Exponential Weighting” or EPEx.

B. Formulation of EPEx

Exponential weighting is a natural way to aggregate the portfolios of different experts. Let E denote the set of individual experts to be pooled together (e.g. UCRP, RMR, etc.). Then, we define the EPEx portfolio to be a convex combination of the individual expert portfolios:

$$\boldsymbol{b}_{\text{EPEx}}(t) = \sum_{i \in E} \rho_i(t) \boldsymbol{b}_i(t) \tag{1}$$

where $\boldsymbol{b}_i(t)$ is the portfolio allocation determined by expert i and $\rho_i(t)$ is the weight assigned to expert i based on expert i's recent performance prior to day t. In particular, $\rho_i(t)$ is given by:

$$\rho_i(0) = \frac{1}{|E|}$$

$$\rho_i(t) \propto \exp(\eta S_i)$$

where η is a parameter controlling the extent to which an expert's recent performance influences the expert's resulting weight. S_i denotes the Sharpe ratio of expert i over a window of time prior to day t. The length of this window must be specified as a parameter. Note that $\rho_i(t)$ must be normalized such that $\sum_{i \in E} \rho_i(t) = 1$, which ensures that the overall EPEx portfolio sums to 1.

Intuitively, an expert's performance during more recent time periods should be a better indicator of how well it will perform on the next day, when compared to older time periods. Following this intuition, we introduce a set of W non-overlapping “performance windows” with lengths $L_1, L_2, ..., L_W$. Each performance window is weighted according to how recently it occurred. Thus, for general values of W, we define the weight for expert i as:

$$\rho_i(t) \propto \sum_{w=0}^{W-1} \alpha^w \exp(\eta S_{i,w}) \tag{2}$$

where $0 < \alpha < 1$ determines the extent to which the influence of older windows is scaled down. $S_{i,w}$ denotes the Sharpe ratio of expert i during window w. Again, note that we must normalize $\rho_i(t)$ such that $\sum_{i \in E} \rho_i(t) = 1$. Also note that $L_i > 3$ such that the Sharpe ratio within each window is well-defined.

At present, EPEx does not explicitly try to optimize in the presence of transaction costs. However, the sliding window nature of EPEx means that the portfolio will probably remain similar on each update as long as the market does not change dramatically. This helps to mitigate the effects of transaction costs.

IV. Experiments

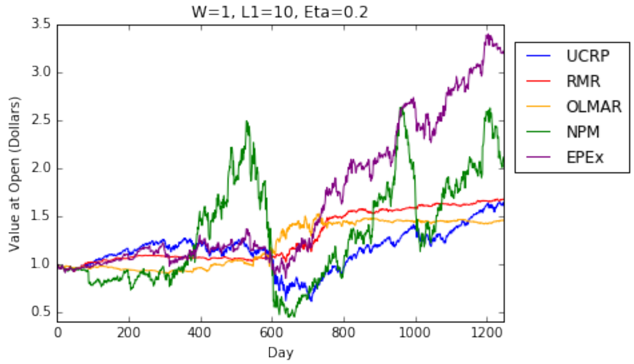
We evaluated our algorithm on a 1247 day period of S&P 500 data. After receiving the opening prices on day t, we ran the UCRP, RMR, OLMAR, and NPM algorithms independently to obtain their desired portfolio allocations for the end of day t. We then performed the update in equation (1) to obtain the net EPEx portfolio based on the weighted portfolios of the 4 underlying experts. We allowed both buying and shorting of stocks, and we enforced a \$0.0005 transaction cost per dollar bought or shorted.

V. Results

Figure 1 shows the total value of assets held by an individual expert running UCRP, RMR, OLMAR, or NPM over the 1247 day period. Figure 1 also shows the value held by an EPEx portfolio consisting of UCRP, RMR, OLMAR, and NPM as its underlying experts. The results in Figure 1 correspond to $\eta = 0.2$ and a single window of length 10 days, i.e. $W = 1$, $L_1 = 10$. The value of α is irrelevant, since $W = 1$.

Table I summarizes the results of Figure 1. As shown in Table I, RMR attains the highest Sharpe ratio. Even though EPEx resulted in roughly twice the mean daily return of RMR, it also incurred roughly 3x the variance in daily returns compared to RMR. This is due to the high variance of the underlying NPM expert.

Figure 1. EPEx Performance vs. Individual Experts



EPEx performance compared to its underlying individual experts. EPEx remains fairly stable during the market crash (approximately day 550) by down-weighting NPM. EPEx is then able to take advantage of the market recovery by up-weighting NPM after the crash.

Table I
EPEX VS. INDIVIDUAL EXPERTS

Algorithm	Annual Return	Sharpe Ratio	Final Dollars
UCRP	0.10	0.35	1.64
RMR	0.10	1.2	1.67
OLMAR	0.08	0.57	1.45
NPM	0.15	0.26	2.11
EPEx	0.24	0.83	3.21

Summary of Figure 1. EPEx attains the highest annual return, but its Sharpe ratio is lower than that of RMR. This is due to the high variance of the underlying NPM portfolio.

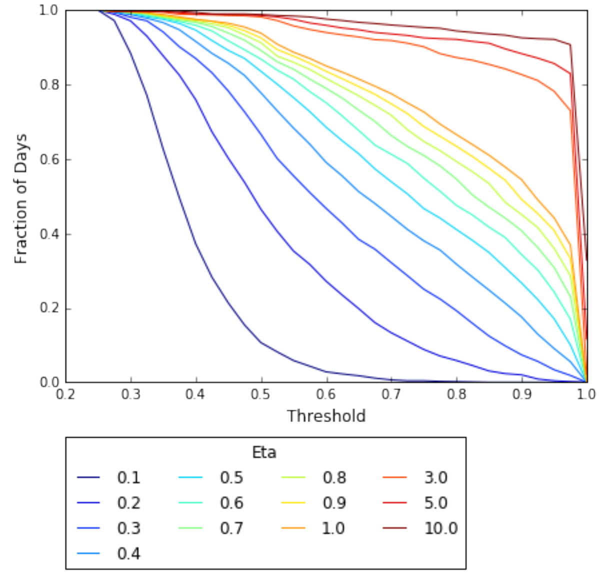
A. Effect of Varying η

The hyperparameter η in (2) can be varied according to the desired weighting effects. When $\eta = 0$, EPEx weights each expert equally regardless of the past performance. Conversely, when η is large, the expert(s) with the worst recent performance will receive little to no weight, while the expert(s) with the best recent performance will receive nearly all of the weight. Figure 2 illustrates this concept by plotting the fraction of days for which any 1 of the experts receives a weight greater than or equal to a given threshold. For example, Figure 2 demonstrates that when $\eta = 10$, a single expert receives 100% of the weight on nearly $1/3$ of the days (the expert need not be the same every day). Figure 2 was generated using the same experimental setup and hyperparameters used for generating Figure 1 (except η).

B. Effect of the Window Length

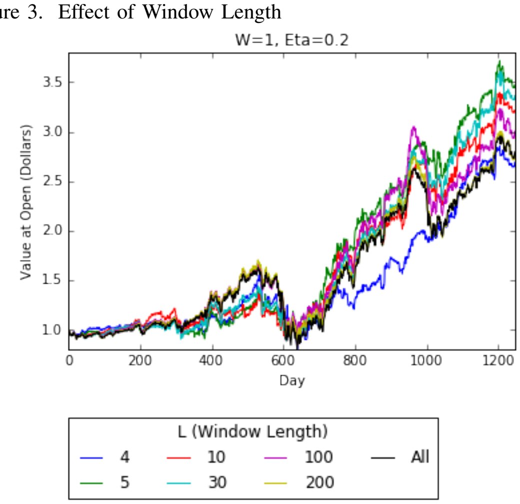
Next we examined the effect of varying the performance window configuration in (2). We repeated the EPEx experiment shown in Figure 1 with serveral different window lengths. The results are shown in Figure 3 and summarized in Table II. We used $\eta = 0.2$ for all window sizes. Again, α is irrelevant since $W = 1$.

Figure 2. Effect of Varying η



Fraction of days for which any 1 of an EPEx portfolio's underlying experts receives a weight greater than or equal to a given threshold. The EPEx portfolio is comprised of UCRP, RMR, OLMAR, and NPM experts. $W = 1$. $L_1 = 10$.

Figure 3. Effect of Window Length



Performance of EPEx for different window sizes ($W = 1$, η=0.2). Underlying experts are UCRP, RMR, OLMAR, and NPM.

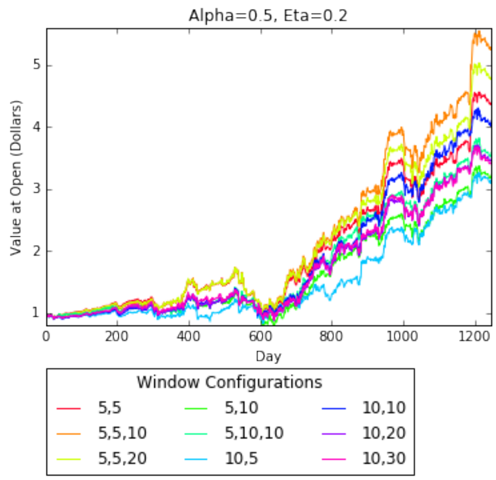
We observe that EPEx with $W = 1$ is not particularly sensitive to the window length. However, performance generally decreases as the window length increases. When EPEx uses a larger window, it reacts more slowly to changes in the market. The exception to the trend is $W = 4$ (the minimum possible window size), which performs significantly worse than $W = 5$. This reflects the high sensitivity of the Sharpe ratio when the window size is small, suggesting that medium-sized windows are preferable.

Table II
SUMMARY OF WINDOW LENGTH EFFECTS

L	Sharpe	Annual Return	Final Wealth (Dollars)
4	0.67	0.20	2.69
5	0.87	0.25	3.45
10	0.83	0.24	3.21
30	0.84	0.25	3.36
100	0.74	0.22	3.00
200	0.67	0.21	2.78
All	0.66	0.21	2.80

Summary of Figure 3. EPEx is not particularly sensitive to the window size when $W = 1$, but performance generally decreases with increasing window length.

Figure 4. Effect of Window Configuration



Performance of EPEx for various window configurations. Legend labels indicate window size configurations. Each configuration is displayed in order from most recent window to oldest window. Each window configuration is contiguous and non-overlapping. $\alpha = 0.5$, $\eta = 0.2$.

C. *Effect of Multiple Windows*

Then we repeated the same experiment shown in Figure 3, except we used window configurations comprised of multiple windows. Here, we set $\alpha = 0.5$. The results of using various window configurations are shown in Figure 4 and summarized in Table III. We observe that the different window configurations result in similar trends. This is primarily because of the fact that the underlying NPM algorithm attains a significant portion of the allocation for all of the configurations beyond day 600 or so. However, (5,5,10) achieved the best performance, particularly in terms of its Sharpe ratio. These results suggest that a small initial window should be chosen, along with small or medium-sized windows after it. This enables EPEx to adapt quickly to a changing market, while avoiding changing its strategy too quickly due to noise.

Table III
SUMMARY OF WINDOW CONFIGURATION EFFECTS

Windows	Sharpe	Annual Return	Final Wealth (Dollars)
5,5	1.04	0.30	4.38
5,5,10	1.56	0.34	5.28
5,5,20	1.09	0.32	4.79
5,10	0.82	0.24	3.23
5,10,10	0.91	0.26	3.59
10,5	0.80	0.23	3.16
10,10	0.98	0.29	4.10
10,20	0.88	0.25	3.49
10,30	0.87	0.25	3.44

Summary of Figure 4. Windows lengths are listed in order from most recent to oldest. For example, "5,10" means the 1st window consists of the previous 5 days, and the 2nd window consists of the 10 days before the start of the 1st window.

VI. CONCLUSIONS AND FUTURE WORK

This initial analysis of EPEx demonstrates its potential for selecting portfolios that perform better than even the best individual expert. We suspect that we could obtain significantly better results by improving the underlying experts and adding more diverse experts. At present, we only employ 3 genuinely distinct experts, as the OLMAR and RMR approaches are relatively similar. In addition to adding more underlying experts to EPEx, we also plan to examine what conditions should be placed on the underlying experts to make them work best under the EPEx framework. For example, some algorithms can be tuned such that they place more emphasis on minimizing risk versus maximizing return. It remains unclear which of these 2 goals should be emphasized when implementing the underlying experts for an EPEx portfolio.

We also plan to experiment more carefully with the EPEx parameters. In particular, we only experimented with $\alpha = 0.5$ so far. Moreover, we plan to develop a more general method for scaling older windows, which will take into account both the size of the window and its distance in the past.

REFERENCES

[1] Li, Bin, and Steven CH Hoi. "Online portfolio selection: A survey." ACM Computing Surveys (CSUR) 46.3 (2014): 35.
[2] Li, Bin, and Steven CH Hoi. "On-line portfolio selection with moving average reversion." arXiv preprint arXiv:1206.4626 (2012).
[3] Huang, Dingjiang, et al. "Robust median reversion strategy for on-line portfolio selection." (2013): 2006.