AMA-K: Aggressive Multi-Temporal Allocation An Algorithm for Aggressive Online Portfolio Selection

Matthew Kruger  
Computer Science and Applied Mathematics  
University of the Witwatersrand  
Johannesburg, South Africa

Terence L. van Zyl  
Institute for Intelligent Systems  
University of Johannesburg  
Johannesburg, South Africa

Andrew Paskaramoorthy  
Computer Science and Applied Mathematics  
University of the Witwatersrand  
Johannesburg, South Africa

Abstract—Online portfolio selection is an integral component of wealth management. The fundamental undertaking is to maximise returns while minimising risk given investor constraints. We aim to examine and improve modern strategies to generate higher returns in a variety of market conditions. By integrating simple data mining, optimisation techniques and machine learning procedures, we aim to generate aggressive and consistent high yield portfolios. This leads to a new methodology of Pattern-Matching that may yield further advances in dynamic and competitive portfolio construction. The resulting strategies outperform a variety of benchmarks, when compared using Maximum Drawdown, Annualised Percentage Yield and Annualised Sharpe Ratio, that make use of similar approaches. The proposed strategy returns showcase acceptable risk with high reward that performs well in a variety of market conditions. We conclude that our algorithm provides an improvement in searching for optimal portfolios compared to existing methods.

Index Terms—Portfolio Selection, Portfolio Optimisation, Data Mining, Clustering, Multi-Temporal

I. INTRODUCTION

Multi-period portfolio choice is a central problem in finance. It is described by an investor who faces the problem of determining how to sequentially allocate his capital to maximise some performance measure over multiple periods. Online portfolio selection algorithms tackle the problem of maximising cumulative wealth by adaptively identifying and exploiting patterns in historical data [1]. The key feature of these algorithms is that they are online: patterns and portfolio decisions update upon the arrival of new data, thereby adapting to changing market conditions [1].

Online portfolio selection algorithms can be classified according to their update scheme. Traditional algorithms forecast asset returns and are used to update the current portfolio. Li et al. (2014) classifies traditional algorithms into the following categories [1]:

- Follow-The-Winner (FTR) algorithms assume that recent stock performance would persist and so transfer capital from the worst-performing stocks to the best-performing stocks.
- Follow-The-Loser (FTL) algorithms assume that recent stock performance would revert to a long-run mean and so transfer capital from the best-performing stocks to the worst-performing stocks.
- Pattern-Matching based (PM) algorithms assume that market conditions repeat themselves and so they allocate capital based on what was optimal for similar historical periods.

The Pattern-Matching based approach has the least restrictive assumption about market behaviour. This affords greater flexibility in algorithm design and allows these algorithms to exploit a wider range of market conditions, thereby outperforming the other approaches [2–3]. In particular, the CORN-K (CORrelation-driven Non-parametric learning) algorithm appears to demonstrate the best results. Recently, the CORN-K algorithm has been extended to incorporate risk in its portfolio selection [4, 5].

However, the CORN-K algorithm (and its extensions) often output a cautious portfolio which restricts its returns. In short, this occurs when the algorithm is unable to detect a subset of historical data that is similar to the recent data and therefore, allocates wealth equally across assets.

To do this, we propose the Aggressive Multi-Temporal Allocation (AMA-K) algorithm, which combines the Pattern-Matching and Follow-the-Winner principles.

The rest of this paper is organised as follows. In Section II we define the portfolio selection problem. Section III discusses the CORN-K algorithm. Section IV introduces our AMA-K algorithm.

II. ONLINE PORTFOLIO SELECTION

An investor wants to allocate his initial capital $S_0$ into a portfolio of $m$ securities for each of the $n$ trading days to maximise his terminal wealth $S_n$. The investor’s portfolios are represented by $b_t = (b(t,1), \ldots, b(t,m))$, where $b(t,j)$ is a proportion of the capital invested in security $j$ at time $t$. Furthermore, portfolio positions are constrained to be non-negative $b(t,j) \geq 0$ and all capital is invested at each period $\sum_{j=1}^{m} b(t,j) = 1$.

Define the price relative for security $j \in 1, \ldots, m$ at day $t \in 1, \ldots, n$ as $x_{t,j} = \frac{P(t,j)}{P(t-1,j)}$, where $P(t,j)$ denotes the log-price. Hence, denote the price relative vector as $x_t = (x_{t,1}, \ldots, x_{t,m}) \in \mathbb{R}_+^m$. A sequence of price relative vectors are used to define a market window $X_t^{w} = (x_{t-w}, x_{t-w+1}, \ldots, x_t)$, where $w$ is the given window size.

An online portfolio selection algorithm $A$ is a function that takes the historical price data at time $t$ and outputs a portfolio:

$$A\left(\bigcup_{i=1}^{t-1} \{X_i^{w}\}\right) = b_t.$$  

(1)
The portfolio is constructed at the start of period $t$, using all information up until then. The terminal wealth at the end of period $n$ is given by:

$$S_n = S_0 \sum_{t=0}^{n-1} (b_t' x_{t+1}).$$ (2)

For tractability, we make the following assumptions: each asset is arbitrarily divisible, desired quantities can be traded at the most recent closing price, and market prices are not affected by the investor’s actions. In addition, we ignore trading costs and do not allow for borrowing or short-selling.

III. CORRELATION-DRIVEN NON-PARAMETRIC LEARNING

CORN-Based strategies use experts that construct portfolios using previous market windows. Experts have a portfolio $b_t$ and have a cumulative wealth $s_t$, at time $t$. $P \times W$ experts are considered, each defined by a window size $w \in (1, \ldots, W)$ and a Pearson product-moment correlation coefficient threshold $\rho \in (\frac{1}{W}, \ldots, \frac{W-1}{W})$ [3]. In top $K$ based strategies, the $K$ experts with the most cumulative wealth at time $t$ have their portfolios combined. Each expert is responsible for $\frac{1}{K}$ wealth of the portfolio’s allocation for a given day. This combined portfolio is the agent’s portfolio at time $t$.

Each expert compares the most recent market window at time $t$ with all historical market windows of the same size. Each expert searches for their optimal portfolio using a given set of data that is equal to or greater than their respective $\rho$. Days that match this required $\rho$ are called correlation similar days and is represented by $C_t(w, p)$ [3]. Experts update their wealth at the end of a day using this portfolio and the day’s returns.

An expert’s portfolio is determined by $b$ that maximises Equation [3] at time $t$:

$$b_t(w, p) = \arg\max_b \prod_{i \in C_t(w, p)} (b \cdot x_i)$$ (3)

At times, the correlation similar set of days may be small or empty. In this case the expert returns uniform portfolio [3–5]. A uniform portfolio is when wealth is equally distributed amongst all assets - which generally have lower returns.

DRICORN-K is a variation of CORN-K that classifies the market and adjusts accordingly. Classification is done through the use of $\beta$ (market Beta) in searching for the optimal portfolio [5]. Utilising $\beta$ allows for more aggressive/defensive portfolios based on current market conditions [5]. At times, $\beta$ does not impact the portfolio construction, in which case DRICORN-K returns a similar portfolio to CORN-K.

IV. METHOD

Clustering previously has been employed in online portfolio selection [6–10]. Similar approaches are given by Khedmati et al. (2020) where portfolios are optimised using clustering techniques, market windows, Pattern-Matching and similar day samples [6, 8]. Nanda et al. (2010) found that K-means clustering provided the best result for online portfolio selection based on cluster compactness using the Bombay Stock Exchange [10]. Our work extends on these previous successes by directly integrating clustering into the CORN-K framework. Further, we introduce a more effective low dimensional representation for market windows that improve the clustering results.

A limitation of CORN-K’s use of correlation similar days is correlation similar days are rare and usually only common amongst experts with smaller window sizes and lower values for $\rho$. Hence, experts that have a suitable quantity of data to produce inference in the market dynamics are unable to do so. To overcome this limitation, we use online K-Means clustering (K$\mu$-online) with Manhattan distance as an alternative to discover sets of cluster similar days. These cluster similar days do not require a correlation coefficient threshold and are considered similar if they belong to the same centroid. Manhattan distance is selected for computational efficiency - alternate metrics could be considered.

Our variation to CORN-K lies in dealing with empty sets of correlation similar days. If we encounter a day where the agent has $C_t(w, p) = \emptyset$ and the market window size is $w > 2$, we make use of our current day’s (day $t$) market vector’s assigned cluster as created in Algorithm [11] We let the correlation similar set be all days assigned to the same cluster and maximise using Equation [3].

A. Aggressive Multi-Temporal Allocation

We have chosen to maintain the method of choosing $K$ best experts in our algorithm. Furthermore, we use the concept of market windows ($X_{t-w}$, for a market window of size $w$ at day $t$), these are matrices that represent consecutive market days’ price relative vectors across all shares.

1) Agent Memory: At the start, all the agent’s experts have $d$ days of market history. As the algorithm proceeds, new days are added to the agent’s memory until it has $2d$ days of market history. At $2d$ days, the agent forgets all but the most recent $d$ days of market history. The choice of $d$ ensures that the agent considers only recent price movements and the length of $2d$ keeps the agent’s portfolio allocations “stable”. A small $d$ value results in an myopic agent that exploits volatility. A high $d$ value results in an agent that looks for “blue chips” that represent long-term growth trends.

Each market window $X_{t-w}$ is represented by a market vector $m_w \in \mathbb{R}^4$ with components: $m_w = [\text{the sum of each stock’s mean in the market window}]$, the market window’s mean, the sum of each stock’s variance and the variance of the market window.

$$\frac{1}{w} \sum_{j=1}^{m} \sum_{k=t-w}^{t} X^{(t,j)}_{(k,j)},$$ (4)

where $j$ represents the $j^{th}$ asset.

$$\frac{1}{w} \sum_{j=1}^{m} \sum_{k=t-w}^{t} \left(X^{(t,j)}_{(k,j)} - \mu_j^2\right),$$ (5)

where $\mu_j$ is the average price relative for the $j^{th}$ asset in the given market window.
We initialise the algorithm with $W = 5$, windows of size $w_i$ where $i \in \{1, \ldots, W\}$ market vectors and $W$ is the maximum window size. $W$'s value is from the original CORN paper [1]. We initialise the number of cluster centroids as $\lfloor \frac{d}{4} \rfloor$ ($d$ represents the number of days). The initial centroid amount was determined using the validation data.

Subsequently, we shift the market window forward by one day and assign $W$ new market vectors for their respective windows $w_i$ to their nearest clusters for that agent’s experts. Noting that market vectors ($w_i$) are assigned to experts with the same window size. We re-do the clustering every $\lfloor \frac{d}{4} \rfloor$ days to keep the allocation of market vectors uniform and relevant. The additional centroids allow for the new unseen market vectors to be represented. Even though this readjustment interval was determined using the validation set, the strategy was found to be insensitive to intervals in the range $\lfloor \frac{d}{4} \rfloor$ to $\lfloor \frac{d}{3} \rfloor$.

Since the $K\mu$-online algorithm does not converge, we terminate the clustering when reassignments affect only $\varepsilon = 0.6\%$ of market vectors. The $0.6\%$ is a manually determined parameter for the general case when the assignment of market vectors to clusters proceeded normally. If the $0.6\%$ threshold is not obtained within ten re-initialisation attempts, we use the last readjustment. This maximum number of attempts allow the clustering process to proceed which may yield undesirable cluster assignments.

Every $2d$ days we reset the $K\mu$-online clustering using the most recent $d$ days of data and $\lfloor \frac{d}{4} \rfloor$ centroids. The reason why we reset every $2d$ days is that it allows the algorithm to provide a balance between factoring in new information and acting less erratically from continuously switching between asset allocations.

2) AMA-$K$: Given that the various (10-day, 120-day and 190-day) agents perform well for their respective time horizons as shown by Table [1], we create an algorithm that combines the agents. This combination creates an agent with three time horizons ($10$-day, 120-day) and long (190-day). Each time horizon has its own set of experts associated to it. The clustering algorithm is repeated for each sub-agent using their $d$ value. Here a sub-agent is an agent over a specific $d$-day time horizon. We take the portfolio for each time horizon and normalize it such that it represents the proportion of the sub-agents’ wealth allocated to each asset at day $t$. These portfolios are merged. The resulting portfolio is then divided by the number of time horizons under consideration (here we divide by three). This resultant portfolio may have a diverse range of assets, which should reduce risk whilst maintaining a high expected return. This idea of efficient diversification is well-founded by Markowitz (1952) in his famous paper Portfolio Selection [11].

B. Metrics

In comparing our algorithm to similar approaches, we use the following metrics that aim to measure performance in a generalised manner.

Maximum Drawdown (MDD) [12]

\[
\text{MDD}(T) = \sup_{\tau \in (0, n)} \left[ \sup_{t \in (0, n)} S(t) - S(\tau) \right]
\]

where $S(\cdot) = \{S_0, S_1, \ldots, S_i\}$.

MDD is a risk evaluation metric which represents the maximum decline from a historical peak of the total wealth($S_i$) achieved at the time $i$. The smaller the MDD value, the more risk tolerant the trading strategy.

Annualised Percentage Yield (APY) [13]

\[
\text{APY}_n = (S_n)^{\frac{1}{y}} - 1
\]

Here $S_n$ is the total return after $n$ trading periods, and $y$ is the number of years corresponding to $n$. APY measures the rate of return that was achieved and it takes into account the effect of compounding. Typically a greater APY is desired.

Annualised Sharpe Ratio (ASR) [14]

\[
\text{SR}_n = \frac{1}{\sigma_p} \left( \text{APY}_n - R_f \right)
\]

Here $\text{SR}_n$ represents the annualised Sharpe Ratio after $n$ periods, $\text{APY}_n$ is an Annualised Percentage Yield (Equation [7]), $R_f$ is the risk-free rate of return and $\sigma_p$ is the annualised standard deviation of daily returns. We use the same assumptions.
as in the DRICORN-K to calculate Equation 5. Where \( R_f \)
set to 4% and \( \sigma_p \) is set to \( \sqrt{252} \) as a result of assuming
an average number of 252 trading days in a given year. The
Sharpe Ratio captures the “return per unit of risk”. A higher
value ASR is preferred.

C. Training and Validation

The data sets used are given in Table I. Assets had their
prices adjusted for dividends and stock splits. Validation
and Testing represent the validation and testing data sets
respectively. All sets are in years - where 252 days is the
average number of trading days in a year. The sets consisted of
6000, 5040 and 2520 days for training, validation and testing
respectively.

In demonstrating memory’s effects, we trained agents using
different values of \( d \) in intervals of ten between 10 and 230
days. Table II is a subset of results for the best performing
sizes for \( d \) in various periods.

| Exchange | Assets | Validation | Test |
|----------|--------|------------|------|
| BIST     | 46     | 2          | 1    |
| BOV      | 28     | 4          | 2    |
| EUR      | 46     | 4          | 2    |
| JSE      | 38     | 4          | 2    |
| NAS      | 41     | 4          | 2    |
| SP5      | 47     | 2          | 1    |

It was observed anecdotally that in markets that experienced
high volatility with the best performing asset constantly chang-
ing, the 10-day agent performed best. In markets that had a
consistent best stock over a long period, the 190-day agent
performed better. The 120-day agent represents a “middle of
the way” agent that yielded an overall better strategy as shown
by the performance across the presented metrics.

D. Testing

Each testing data set consists of one year of data, with
300 days prior for CORN-based strategies to train with.
This extra data is to allow CORN-based strategies to have
a more fair comparison against AMA-K. The CORN-based
algorithms were tuned with their optimal hyper-parameters as
set out in their respective papers [3–5]. In the case of our
implementation, we have segmented \( d \) days for each algorithm,
where \( d \) is the size of the largest sub-agent’s memory - here
this would be 190 days. We will compare our approaches to
some common baselines such as UBAH, CRP and Best

Stock [1]. We also compare our method to EG (Exponential
Gradient) as a showcase of a Follow-The-Winner strategy [15].
In EG we have set \( \eta \) to be 0.05.

V. RESULTS AND DISCUSSION

In the Tables III and IV the number next to a stock
exchange represents which data set it came from. The mean
(\( \mu \)) is the average for the metric across each of the markets
and \( \sigma \) is the standard deviation.

A. Individual Metric Performance

We examine the performances of the various strategies based
on metrics given by Equations 6 and 7 and 8.

1) MDD: As seen in Table III, the best performing strate-
gies are EG and AMA-K. EG outperforms AMA-K by a
considerable margin on average. It should be noted that AMA-
K achieves a low standard deviation for its MDD. Despite this
lower standard deviation, AMA-K had the greatest MDD value
hence, it represents the riskiest strategy. This is expected given
that AMA-K is extremely aggressive in terms of portfolio
allocation. Furthermore, we note that in the two markets that
have periods of flatness, we see that AMA-K performed the
worst. AMA-K received an MDD value that is twice and thrice
greater than the other strategies (in general) for NAS-1 and
JSE-2 respectively.

2) APY: In Table IV we see that AMA-K had the best per-
formance, with AMA-K achieving in the worst case (on NAS-
1) 38.87% better annualised percentage yield for the period
compared to the second-best. AMA-K had the highest mean
APY, however it should be noted that its standard deviation is
fairly large. The second-best strategies are CORN-K and
DRICORN-K. These strategies had nearly half the mean APY
of AMA-K. AMA-K’s better performance to CORN-K and
DRICORN-K results from AMA-K searching for portfolios in
days that these CORN-based strategies’ experts would have
returned uniform portfolios. AMA-K’s strategy has turned
enough of these days into profitable trading days - as reflected
by its mean APY.

3) ASR: Based on Table V we can see that the best per-
forming algorithm is AMA-K. The performance benefits of
AMA-K can be seen in the example of JSE-1, where AMA-
K performs 4 times better than the second-best algorithm in
JSE-1. Therefore, we can conclude that per unit of risk with
a given risk-free rate of 4%, the AMA-K algorithm returns
showcase the risk to reward trade-off at play.

B. General Performance

Looking at the cumulative return of the algorithm on various
markets, we see that the approaches have different patterns
in general. For example for NAS-1 in Figure 4, the general
market trend is upwards, with AMA-K performing the best.
The second-best algorithm is CORN-K and DRICORN-K
which have performed the same. For the first 100 days AMA-
K’s MDD risks are prevalent and AMA-K does the worst.
AMA-K fluctuates heavily in this market and from a bottom
at day 68 where AMA-K lost 7.8% of its total wealth in four
days to being the best performing algorithm at the end.
In general our approach is competitive and leverages CORN-based strategies to produce further gains by searching for optimal portfolios on days that CORN-based strategies would not. Although the risk presented by our algorithm as shown in Figure 4 can be significant, it is an example of a risk to reward trade-off.

VI. CONCLUSION AND FURTHER DEVELOPMENT

Our approach can generate high returns using our memory-based method. The approaches can be volatile and merging them typically results in a more stable strategy (as shown in Tables 6, 7, 8) at the cost of reducing returns of the best agents. Here are possible areas to develop further: Changing or penalise volatility - for lower-risk strategies. Doing a fine-grained grid search for the optimal portfolio to include regret or niche financial metrics should be conducted to further understand the performance of the method in comparison to other modern approaches.

REFERENCES

[1] B. H. Li and S. C.H, “Online portfolio selection: A survey,” ACM Computing Surveys, vol. 46, 2014.

[2] L. Györfi, G. Lugosi, and F. Udina, “Nonparametric kernel-based sequential investment strategies,” Mathematical Finance, vol. 16, pp. 337–357, 2006.
Fig. 1. Cumulative returns BOV-2

Fig. 2. Cumulative returns BIST

Fig. 3. Internal Agents and AMA-K Cumulative returns JSE-2

Fig. 4. Cumulative returns NAS-1

[3] B. H. Li, S. C. H, and V. Gopalkrishnan, “Corn: Correlation-driven non-parametric learning approach for portfolio selection,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 21–29, 2011.

[4] Y. Wang, D. Wang, and T. Zheng, “Racorn-k: risk-aversion pattern matching-based portfolio selection,” Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1816–1820, 2018.

[5] S. Sooklal, T. van Zyl, and A. Paskaramoorthy, “Dricorn-k: A dynamic risk correlation-driven non-parametric algorithm for online portfolio selection,” Proceedings of the First Southern African Conference for AI Research, pp. 183–196, 2020.

[6] S.-H. Liao, H. hui Ho, and H. wen Lin, “Mining stock category association and cluster on taiwan stock market,” Expert Systems with Applications, vol. 35, no. 1, pp. 19–29, 2008.

[7] S. K. Kumari, P. Kumar, J. Priya, S. Surya, and A. K. Bhurjee, “Mean-value at risk portfolio selection problem using clustering technique : A case study,” AIP Conference Proceedings, vol. 2112, no. 1, p. 020178, 2019.

[8] M. Khedmati and P. Azin, “An online portfolio selection algorithm using clustering approaches and considering transaction costs,” Expert Systems with Applications, vol. 159, pp. 113546, 2020.

[9] P. Zuccolotto and G. De Luca, “Dynamic tail dependence clustering of financial time series,” Statistical Papers, vol. 58, 09 2017.

[10] S. Nanda, B. Mahanty, and M. Tiwari, “Clustering indian stock market data for portfolio management,” Expert Systems with Applications, vol. 37, no. 12, pp. 8793–8798, 2010.

[11] H. Markowitz, “Portfolio selection,” The Journal of Finance, vol. 7, pp. 77–91, 1952.

[12] M. Magdon-Ismail and A. Aitiya, “Maximum drawdown,” Risk Magazine, vol. 10, pp. 99–102, 2004.

[13] K. Elton, J. Gruber, and J. Brown, Modern Portfolio Theory and Investment Analysis. J. Wiley & Sons, 2003.

[14] W. Sharpe, “The sharpe ratio,” The Journal of Portfolio Management, vol. 21, no. 1, pp. 49–58, 1994.

[15] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth, “On-line portfolio selection using multiplicative updates,” Mathematical Finance, vol. 8, pp. 325–347, 1998.