석사 학위논문

Master's Thesis

베이지안 평균-중간값 복귀에 기반한 온라인 포트폴리오 선택전략

Bayesian Ensembled Mean-Median Reversion based Strategy for On-line Portfolio Selection

쿠말 아비쉑 (Kumar, Abhishek)

지식서비스공학과

Department of Knowledge Service Engineering

KAIST

2016

베이지안 평균-중간값 복귀에 기반한 온라인 포트폴리오 선택전략

Bayesian Ensembled Mean-Median Reversion based Strategy for On-line Portfolio Selection

Bayesian Ensembled Mean-Median Reversion based Strategy for On-line Portfolio Selection

Advisor : Professor Segev, Aviv

by

Kumar, Abhishek
Department of Knowledge Service Engineering
KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Knowledge Service Engineering . The study was conducted in accordance with Code of Research Ethics[1].

2016. 06. 03.

Approved by

Professor Segev, Aviv

[Advisor]

---

[1]Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

베이지안 평균-중간값 복귀에 기반한 온라인 포트폴리오 선택전략

쿠말 아비쉑

위 논문은 한국과학기술원 석사학위논문으로
학위논문심사위원회에서 심사 통과하였음.

2016년 06월 03일

심사위원장	Segev, Aviv	(인)
심사위원	Yi, Mun Yong	(인)
심사위원	Lee, Uichin	(인)

MKSE 20144644

쿠말 아비쉑. Kumar, Abhishek. Bayesian Ensembled Mean-Median Reversion based Strategy for On-line Portfolio Selection. 베이지안 평균-중간값 복귀에 기반한 온라인 포트폴리오 선택전략. Department of Knowledge Service Engineering . 2016. 40p. Advisor Prof. Segev, Aviv. Text in English.

ABSTRACT

Online portfolio selection, one of the major fundamental problems in finance, has been explored quite extensively in recent years by machine learning and artificial intelligence communities. Recent state-of-the-art methods have focused on *Mean Reversion* significantly and have demonstrated outstanding performance. Another version of the same phenomenon, *Median Reversion* has also performed well and demonstrated its ability to be robust against noises and outliers. Another important characteristic is *Momentum*. In this paper, Bayesian ensembling approach to exploit both *Mean Reversion* and *Median Reversion* simultaneously based on momentum associated with each one, has been proposed for on-line portfolio selection task. The proposed method demonstrates its effectiveness by outperforming current state-of-the-art algorithms on several datasets.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Portfolio selection is one of major fundamental problems of finance which deals with allocation of the wealth to a set of assets which constitute the portfolio so that the investor can attain certain financial objectives over a certain period of time. Now this problem is no longer studied in finance only, but is also studied in fields like mathematics, machine learning and artificial intelligence [1]. This problem is generally studied in light of two mathematical models: Mean-variance Theory [14] and Kelly investment [15, 2, 3]. Mean-variance theory deals with trade-off between return (or mean) and risk (or variance). It is mainly suitable for single-period portfolio selection. Kelly investment is quite suitable for multi-period portfolio selection. Unlike any kind of trade-off, Kelly investment simply focuses on maximizing the expected log return of the portfolio. Most of recent works in this field have exploited the concept of Kelly investment, especially on-line portfolio selection task.

One of the most significant characteristics in finance is the *Reversion*. It has been studied quite extensively in recent years [30, 21, 57, 62] in light of on-line portfolio selection task. These algorithms based on *mean reversion* [30, 21, 57, 62] have performed quite well on real datasets. Another variant of this property is *median reversion* [20]. Using *median reversion* is an ideal choice in presence of noise and outliers. But if the dataset is relatively clean, using *mean reversion* is a better choice. Another important characteristics is *Momentum*. So, it leads to problems: 1) How to exploit *Reversion* and *Momentum* 2) When to exploit *mean reversion* property and when to exploit *median reversion* property.

To address the above problems, this experiment proposes a different approach which exploits both *mean reversion* and *median reversion*, named "Bayesian Ensembled Mean-Median Reversion based Strategy for On-line Portfolio Selection", abbreviated as BMMR. The basic idea is to ensemble both properties using Bayesian probability, then to learn from mistake made in previous prediction followed by explicit estimation of the portfolio using on-line machine learning approach [51]. The proposed method outperforms current state-of-the-art methods on many real datasets in terms of the final *cumulative wealth*. Also, on other datasets it makes sure that anyone who makes investment based on this algorithm, will have higher final cumulative wealth than the one following market, index fund and many other algorithms [13, 30, 11, 8]. Also, running time of BMMR is quite efficient. It is quite suitable for high frequency trading applications [4, 5].

The remainder of the paper is organized as follows: Chapter 2 describes the on-line portfolio selection setting. Chapter 3 reviews some recent works done in on-line portfolio selection from perspective of computer science. Chapter 4 describes the proposed BMMR approach in detail. Chapter 5 covers experiments and comparative studies along with metrics used to measure the performance and the datasets used to test the performance. Finally, chapter 6 offers some concluding remarks.

Chapter 2. Problem Settings

A portfolio of m assets has been considered for the purpose of investment in financial market. This investment lasts for n trading periods. On the t^{th} trading period, the price-relative vector for the m assets of the portfolio is represented by $x_t = (x_{t,1},\ x_{t,2},......,\ x_{t,m})$ where element $x_{t,i}$ represents price-relative of asset i on period t and $x_{t,i}$ is formulated as ratio of the closing price of the asset i on period t to the closing price of the asset i on period *t-1* i.e. $\frac{p_{t,i}}{p_{t-1,i}}$. Let $x_1^n = (x_1,\ x_2,......,\ x_n)$ be represent the sequence of m dimensional price-relative vector for n periods starting from period 1 i.e. beginning.

The investment in the portfolio on the t^{th} trading period is represented by *portfolio weight vector* $b_t = (b_{t,1},\ b_{t,2},......,\ b_{t,m})$. Each element $b_{t,i}$ represents proportion of the wealth invested in the asset i on the t^{th} trading period. The portfolio is assumed to be self financed i.e. it requires finance only at the beginning of the investment period. It also assumes no margin or short selling. Therefore, each element of portfolio vector is non-negative and all element sums up to one i.e. $b_t \in \Delta_m$, where $\Delta_m = \{b_t : b_t \in R_+^m,\ \sum_{j=1}^m b_{t,j} = 1\}$. The investment procedure is represented by the *portfolio selection strategy.* This *portfolio selection strategy* is given as: $b_1 = \{\frac{1}{m},\ ....,\ \frac{1}{m}\}$ and a sequence of mappings $R_+^{m(t-1)} \rightarrow \Delta_m$, t = 2,3,..., where $b_t = b_t(x_1,\ x_2,......,\ x_{t-1})$ is the portfolio invested on the t^{th} trading period having knowledge of past price-relatives of assets of the portfolio $x^{t-1} = \{x_1,\ x_2,......,\ x_{t-1}\}$. The *portfolio selection strategy* for n trading period is given by $b^n = (b_1,\ ...,\ b_n)$.

For the t^{th} trading period, an investment formulated by portfolio b_t leads to portfolio daily return m_t, i.e., on the t^{th} trading period, the wealth increases by the factor of c_t which is given by $b_t^T x_t = \sum_{i=1}^m b_{t,i} x_{t,i}$. Evey next trading period, we reinvest entire wealth accumulated so far and use price-relatives. It leads to multiplicative cumulative return of investments. Therefore, after n trading periods, the investment formulated by portfolio selection strategy b^n results in cumulative portfolio wealth C_n. By n^{th} trading period, the initial wealth increases by factor of $\prod_{t=1}^n b_t^T x_t$. Therefore, the cumulative wealth on n^{th} trading period is given by, $C_n(b_1^n, x_1^n) = C_0 \prod_{t=1}^n (b_t^T x_t)$ where C_0 is the initial wealth i.e. amount of wealth invested to create portfolio in the very beginning. In this experiment, the initial wealth C_0 has set to 1.

Now, the problem, i.e. on-line Portfolio Selection, is dealt as sequential decision making. The goal of investor is to formulate investment strategy b_1^n such that the final portfolio cumulative wealth C_n is maximum. In the beginning of each t^{th} trading period, investor has access to historical daily price-relatives till last trading period i.e. $t-1$. The investor does not know price relative information of portfolio assets because market has not yet revealed the actual price. The paper assumes that market reveals actual price of assets at the end of the period in form of closing price. So, the investor uses the available information upto $(t-1)^{th}$ trading period and formulates strategy b_t for next daily price-relative x_t. The investor keeps repeating this procedure this until the end of trading period. Finally, the investment strategy is judged based on final cumulative wealth C_n. Algorithm 1 summarizes the problem setting.

ALGORITHM 1: On-line Portfolio Selection Setting

Input: Initial Wealth $C_0 = 1$, Initial Investment Strategy $b_1 =\{ \frac{1}{m}, ..., \frac{1}{m}\}$, Total number of trading periods N

Output: Final Cumulative Wealth C_n

Initialization $t = 1$

while $t \leq N$ **do**

Investor formulates portfolio b_t based on historical price information (b^{t-1}, x^{t-1})

Actual price-relative on trading period t is generated by the market i.e. x_t

Portfolio daily return is given by $s_t = b_t^T x_t$ and the cumulative wealth is updated to $C_t = C_{t-1} \times b_t^T x_t$

Based on score of investment on t^{th} trading period, the investor updates its portfolio selection rules

$t = t + 1$

end

In order to define the problem as per in Algorithm 1, apart from no margin/short selling, certain assumptions have been made.

1. No Transaction Cost: Although every time any change like buy/sell to any asset of portfolio is made in the market, it incurs certain amount of transaction fee. The model does not consider transaction into account.

2. No Tax: The model assumes no tax.

3. High Market Liquidity: The model allows investor to buy and sell desired quantities of portfolio asset at last closing price.

4. Impact Cost: The model assumes that portfolio selection strategy suggested by it does not bring major abruptions in the market.

Chapter 3. Related Work

In this chapter, various strategies used in trading for portfolio selection are discussed. All such trading strategies can be into one of five categories as discussed below:

1. **Benchmarks**
2. **Pattern Matching**
3. **Meta-Learning**
4. **Follow the Winner**
5. **Follow the Leader**

Now, some of benchmarks and state-of-the-art algorithms (or trading strategies) in each category are discussed.

3.1 Benchmarks

There are three baselines for any trading strategy for portfolio selection. The most common one is *Buy & Hold* strategy (*BAH*). Under this strategy, the investor buys certain quantities of all assets of the portfolio at the beginning of the investment. The investor holds its initial investment b_1 throughout the duration of investment. If the investor allocates its wealth uniformly among all assets of the portfolio, i.e. $b_1 = \{\frac{1}{m}, ..., \frac{1}{m}\}$ and holds this portfolio throughout the investment period, it is called *Uniform BAH* strategy. In this experiment, *Uniform BAH* strategy is considered as the *Market* strategy because it simply follows the market.

The other baseline *Constant Rebalanced Portfolios* strategy *CRP*. It was introduced by [6]. It is considered as one of the classical trading strategies in finance. In contrary to the *BAH* strategy, it makes changes to the proportion of wealth accumulated so far, into different assets of the portfolio. So, using this strategy, the final cumulative wealth of the investor after n trading period becomes $C_n = C_0 \prod_{t=1}^{n}(b_t^T x_t)$ where C_0 is the initial wealth. One variant of this strategy is *Unform CRP (UCRP)* where initial portfolio b_1 is uniformly distributed. Other variation of *CRP* is the best *Best CRP* Strategy *BCRP*. It is hindsight strategy but quite often, it is used for as baseline when dealing with any other trading strategy. The wealth obtained using this strategy in hindsight after n trading periods is $C_n^* = max_{b_t \in \Delta_m} C_n$. This experiment also uses it as the benchmark.

Another widely adopted baseline strategy is the *Best Stock* strategy. Under this strategy, the investor invests all of his asset on one stock which is the best performing one in the hindsight. Mathematically, the final wealth which could be accumulated using this strategy, can be expressed in terms of *Buy & Hold* strategy as $C_n = C_n(BAH(b_1^*))$ where b_1^* is the initial investment calculated in the hindsight, which puts all wealth on the best performing stock.

3.2 Pattern Matching

History repeats itself is one of major characteristics of financial market. It is one of founding principles of *Technical Analysis* which is heavily used to forecast the direction of prices through the study of historical market data, especially price and volume [7].

All pattern matching algorithms for portfolio selection involve two major steps: 1) selection of historically similar sample 2) portfolio optimization [8]. Such algorithms don't need any parameter for predicting price-relatives of next trading period. Therefore, they are also called nonparametric predicting algorithms [9]. There are several nonparametric algorithms for on-line portfolio selection[8, 11, 17, 13].

3.2.1 Selection of historically similar samples

This step basically implies choosing market window of historical price-relatives with the current market window of price-relatives. First, the current window of size w is chosen, then it is iterated from $i = w + 1, ..., t$ to find a set of all similar windows. All these similar windows are assigned probability based on degree of similarity.

Nonparametric *histogram based* technique for similar sample selection uses a set of discretized partitions, and partitions both market windows, and chooses similar price-relatives [9].

Nonparametric *kernel* based technique uses euclidean distance to find historically similar price-relative windows to the current one [8]. The set of similar historical similar windows can be defined as:

$$C_H(x_1^t, w) = \{w < i < t + 1 : \|x_{t-w+1}^t - x_{i-1}^{t-1}\| \leq \tfrac{c}{l}\}$$

where c and l are threshold variables which control number of similar windows. According to authors [8], two variables were used for the purpose of theoretical analysis .

Nonparametric *nearest neighbor* based technique for sample selection finds historical prince relative windows which are among the l nearest neighbors of the current window [11]. The set of similar historical similar windows can be defined as:

$$C_H(x_1^t, w) = \{w < i < t + 1 : x_{i-1}^{t-1} \text{ is within the } l \text{ nearest neighborhood of } x_{t-w+1}^t\}$$

where l is the threshold variable.

Correlation driven nonparametric technique for sample selection uses Pearson Correlation coefficient [12] to exploit linear similarity between current window and historically similar windows [13]. The set of similar historical similar windows can be defined as:

$$C_H(x_1^t, w) = \{w < i < t + 1 : pearson\_correlation(x_{i-w}^{i-1}, x_{t-w+1}^t) \geq l\}$$

where l is the correlation coefficient threshold variable.

3.2.2 Portfolio optimization techniques

As mentioned above, the second step among pattern matching algorithms is to construct the optimal portfolio using set of historically similar windows. Portfolio optimization can be done using two approaches: Markowitz's Mean Variance Theory [14] and Kelly's Capital Growth Theory [15, 16].

Recently, *log-optimal* portfolio technique was introduced for optimization based on set of historically similar price-relative windows based on the Kelly's Growth Optimal Theory [8]. The log-optimal utility function is defined as:

$$U_L(b, C(x_1^t)) = E\{\log b{\cdot}x|x_i, i\epsilon C(x_1^t)\} = \textstyle\sum_{i\epsilon C(x_1^t)} P_i \log b \cdot x_i$$

where P_i is probability weight of i^{th} similar window. In [8], uniform probability was assigned to all historically similar windows. The log-optimal utility in this case, can be expressed as:

$$U_L(b, C(x_1^t)) = \textstyle\sum_{i\epsilon C(x_1^t)} \log b \cdot x_i$$

There are also several other methods besides log-optimal portfolio methods: *Semi-log optimal* portfolio [17, 18], *GV-type* portfolio [22] and *Markowitz-type* portfolio [23].

3.3 Meta-Learning

On-line portfolio selection based on the *Meta learning* approach [32] are similar to expert learning [24] widely used in field of machine learning. A **Meta-Learning** algorithm consists of several base experts. Each expert may have same strategy or different strategy. Each expert individually generates investment strategy b_{t+1} for the next trading period $t+1$. Afterwards, *Meta-Learning* algorithm combines all experts' output to generate final optimal investment strategy b^*_{t+1}. Some of widely used meta learning algorithms for on-line portfolio selection are as follows:

3.3.1 Aggregating Algorithms

Aggregating Algorithms (AA) [24, 25] aggregate underlying base experts based on Bayesian merging approach. The weights of underlying experts using AA are updated as:

$$P_{t+1}(A) = \int_A \beta^{l(x_t, \gamma_t(\theta))} P_t(d\theta)$$

where A is set of underlying experts, $l(x, \gamma)$ is loss function, $\gamma_t(\theta)$ is action by expert θ at time t, P_t is weights of the experts at time t and $\beta = e^{-\eta}$ where η is positive learning rate.

3.3.2 Fast Universalization

Fast Universalization (FU) has been proposed by Akcoglu et al.[27, 28]. It is extension of Cover's Universal Portfolio [29]. It provides investment strategies for a single stock as well as a portfolio of stocks. FU distributes the wealth among underlying base experts. These base experts work independently and final cumulative wealth is generated by collecting wealth generated by these base experts. Asymptotically it accumulates wealth equal to an optimal fixed convex combination of base experts.

Some FU trading algorithms also use *Buy & Hold* strategy. Borodin et al. [30, 31] uses *Buy & Hold* strategy to combine underlying Anticor experts.

3.3.3 on-line Gradient & Newton Updates

on-line Gradient Update (OGU) algorithm and *on-line Newton Update* (ONU) was proposed by Das et al. [32]. These two algorithms are meta optimization algorithms based on *Exponential Gradient*(EG) [33] and *on-line Newton Step* (ONS) [34] respectively. Theoretical growth rate of OGU and ONS is similar to the one achieved by the optimal convex combination of the underlying experts. If any of underlying expert is universal, the whole algorithm has property of universality.

3.3.4 Follow the Leading History

Follow the Leading History (FLH) algorithm for portfolio selection was proposed by Hazan el al. [35]. It works well in constantly changing environments. FLH combines several universal base experts i.e. ONS algorithm. FLH maintains a set of dynamic experts. High performing experts participate and low performing ones don't. High performing experts decide proportion of the wealth to each asset of the portfolio using a meta-learning algorithm such as Herbster-Warmuch algorithm [36]. Such meta-learning algorithm tracks the best experts. Theoretical performance of combination of FLH and ONS is better than one through simply ONS.

3.4 Follow the Winner

Trading algorithms under this category are driven by **One which performs well today, also performs well tomorrow** characteristic of financial market. So, such trading algorithm puts more capital on those assets/experts of the portfolio which performed successfully on the previous period (or day). Such algorithms often use BCRP strategy instead of simply following benchmarks approaches like following market or following the best stock. There are two major algorithms following this approach.

3.4.1 Universal Portfolios

Universal Portfolios (UP) [29] updates the portfolio through historical performance weighted average of all possible constant rebalanced portfolios in the entire simplex domain. UP's investment strategy on any trading period is calculated as:

Initial Investment Strategy, $b_1 = (\frac{1}{m}, ..., \frac{1}{m})$

Investment Strategy on $(t+1)^{th}$ trading period, $b_{t+1} = \frac{\int_{\Delta_m} \mathbf{b} C_t(\mathbf{b}) d\mathbf{b}}{\int_{\Delta_m} C_t(\mathbf{b}) d\mathbf{b}}$

From these two mathematical equations, it can be said that the final cumulative wealth accumulated by the UP is unform weighted average of all CRP experts' wealth, $C_n(UP) = \int_{\Delta_m} C_t(\mathbf{b}) d\mathbf{b}$.

3.4.2 Exponential Gradient

Exponential Gradient (EG) [33] is based on algorithm proposed by [37] for mixture estimation. The optimization step for this algorithm can be exprssed as follows:

$$b_{t+1} = \underset{b \in \Delta_m}{\operatorname{argmax}} \{\eta \log b \cdot x_t - R(b, b_t)\}$$

where, $R(b, b_t)$ is for regularization and η is positive learning rate. The central idea in EG's optimization step is to find the last best performing asset and use regularization to keep other information of the portfolio.

3.4.3 Stochastic Optimization

Successive Rebalanced Portfolios (SCRP) [39] works well for stationary market. It uses BCRP updating technique fo formulate investment strategy on any next trading period $t+1^{th}$:

$$b_{t+1} = \underset{b \in \Delta_m}{\operatorname{argmax}} \sum_{\tau=1}^{t} \log(b \cdot x_\tau)$$

The paper [38] uses stochastic optimization to find the optimal investment strategy. It leads to SCRP's portfolio updates [39]. *Adaptive Portfolio Selection* (APS) proposed by Gaivoronski [40] performs three portfolio tasks: adaptive Markowitz portfolio, log-optimal CRP and index tracking.

on-line Newton Step [34] uses L2-norm regularization through on-line convex optimization to retrieve optimal investment strategy [41, 42, 43, 44]. It's investment stategy can be expressed as:

$$b_{t+1} = \underset{b \in \Delta_m}{\operatorname{argmax}} \sum_{\tau=1}^{t} \log\left(b \cdot x_\tau\right) - \frac{\beta}{2} R(b)$$

where R(b) is regularization term which contains information about the next investment strategy (or portfolio) and β is trade-off parameter.

3.5 Follow the Loser

Algorithms under this category transfers wealth from high performing assets to low performing assets. Asymptotic growth rate of *follow the winner* algorithms is same as that of *BCRP*. But in general, the final cumulative wealth generated by *follow the winner* algorithms is less than achieved by the *BCRP*. Moreover, In order for *BCRP* strategy to work, asset returns must be independent and identically distributed random variables (i.i.d.). But according to few financial studies [45, 46], real asset returns in the financial market may not be i.i.d. making BCRP not the best choice for investment. *Follow the loser* algorithms assume that asset returns contain property of *mean reversion* [47, 48, 49].

3.5.1 Anti Correlation

Anti Correlation (Anticor) proposed by Borodin et. al [30] is based on *follow the loser* principle. Anticor makes statistical bets on the consistency of positive lagged cross-correlation and negative cross-correlation in order to exploit mean reversion property. Anticor calculates cross-correlation between two specific market windows and transfers wealth from high performing assets to low performing ones.

3.5.2 Passive Aggressive Mean Reversion

Passive Aggressive Mean Reversion algorithm proposed by Li et al. [21] exploits mean reversion propoerty using an on-line machine learning algorithm: Passive Aggressive (PA) on-line learning [51, 52]. PAMR involves design of loss function $l(b; x_t)$ to exploit mean reversion property. In summary, portfolio optimization using PAMR can be expressed as follows:

$$b_{t+1} = \underset{b \in \Delta_m}{\operatorname{argmin}} \tfrac{1}{2} \|b - b_t\|^2 \text{ such that } l(b; x_t) = 0$$

Solving this optimization expression leads to :

$$b_{t+1} = b_t - \tau_t(x_t - \bar{x}_t 1) \text{ where } \tau = \max\{0, \tfrac{b_t \cdot x_t - \epsilon}{\|x_t - \bar{x}_t 1\|^2}\}$$

Afterwards, simplex projection is applied to enforce non negativity constraints on wealth allocation to assets of then portfolio [63].

3.5.3 Confidence Weighted Mean Reversion

Most of algorithms for portfolio optimization only exploit first under information of the portfolio, but *Confidence Weighted Mean Reversion* (CWMR) [57] algorithm exploits second order of information too using confidence weighted on-line learning [58, 59, 60, 61]. According to the CWMR, the portfolio vector follows multivariate Gaussian distribution. Mean $\mu \in R^{m \times n}$ of CWMR portfolio vector represents the knowledge of the CWMR portfolio. Nonzero diagonal elements of CWMR covariance matrix $\Sigma \in R^{m \times n}$ represents confidence in the corresponding portfolio vector. The CWMR updates mean and covariance matrix. Mathematically, CWMR is expressed as follows:

$$(\mu_{t+1}, \Sigma_{t+1}) = \underset{\mu \in \Delta_m, \Sigma}{\operatorname{argmin}} \mathcal{D}_{KL}(\mathcal{N}(\mu, \Sigma) || \mathcal{N}(\mu_t, \Sigma_t)) \text{ such that } \mathit{Probability}[\mu \cdot x_t \leq \epsilon] \geq \theta$$

CWMR keeps the next distribution close to the last distribution in terms of Kullback-Leibler divergence if the probability of a portfolio return lower than ϵ is higher than a specified threshold θ.

3.5.4 On-line Moving Average Mean Reversion

The last three *follow the loser* algorithms follow single period mean reversion. But in general, it is not satisfied. It is empirically shown that it does not work on DJIA dataset [21, 57]. *on-line Moving Avergae Reversion* (OLMAR) [62] defines multi-period mean reversion to predict price-relatives of next trading period. So, the price-relative according to the simple moving average reversion is next price-relative vector is simply moving average of price-relatives within the window. Mathematically, it can be expressed as follows:

$$\tilde{x}_{t+1}(w) = \frac{MovingAverage_t(w)}{p_t} = \frac{1}{w}(1 + \frac{1}{x_t} + ... + \frac{1}{\odot_{i=0}^{w-2} x_{t-i}})$$

where $\odot$ is element wise product and w is window size. After predicting price-relatives, it uses Passive Aggressive on-line Learning [51] for portfolio optimization:

$$b_{t+1} = \underset{b \in \Delta_m, \Sigma}{\operatorname{argmin}} \frac{1}{2}\|b - b_t\|^2 \text{ such that } b \cdot \hat{x}_{t+1} \geq \epsilon$$

3.5.5 Robust Median Reversion

The previous method *OLMAR* works well if there are no major noises or outliers in price-relative dataset. But financial data is highly complex. There is no perfect model to describe exact behavior of market through the data only because it contains too many noises and outliers. So, *OLMAR* suffers from estimation error. To make the prediction more robust against these noises and outliers, Huang et al. [20] proposed *Robust Median Reversion* method to predict price-relatives for next trading period. It uses L_1 median estimator to estimate price-relatives for next trading period as follows [53, 54, 55]:

$$\tilde{x}_{t+1} = \frac{\mu_{t+1}}{p_t} = \frac{\underset{\mu}{\operatorname{argmin}} \sum_{i=0}^{w-1} \|p_{t-i} - \mu\|}{p_t}$$

where μ_{t+1} is estimated price for the $(t+1)^{th}$ trading period and w is number of latest trading periods used to estimate price for next trading period. After predicting price-relatives, it uses Passive Aggressive On-line Learning [51] for portfolio optimization:

$$b_{t+1} = \underset{b \in \Delta_m, \Sigma}{\operatorname{argmin}} \frac{1}{2}\|b - b_t\|^2 \text{ such that } b \cdot \hat{x}_{t+1} \geq \epsilon$$

Chapter 4. Method

4.1 Bayesian Ensembling

4.1.1 Motivation

Financial data are highly complex in nature. It contains multiple characteristics. There are three major long-term characteristics: *reversion*, *momentum* and *repetition of history.* Empirically, exploiting *reversion* leads to superior performance in terms of final cumulative wealth. Most of existing state-of-the-art methods focus on exploiting only one characteristics. The main hypothesis for the proposed method is that exploiting more characteristics may lead to more accurate prediction of price information and ultimately better performance. So, the proposed method tries to exploit *reversion* and *momentum.*

These two characteristics are opposite in nature. *Reversion* characteristic is also known as *Follow the Loser* approach. *Momentum* characteristic is also known as *Follow the Winner* approach. Using these two characteristics means, to follow the winning stocks as well as the losing stocks at the same time, which is a dilemma in itself. So, the proposed method tries to exploit *reversion* characteristic more efficiently. *Reversion* is exploited by either *mean* or *median. Mean* is a better choice on clean datasets. *Median* is a better choice on datasets containing noises or outliers. But the real-world dataset is mixed, i.e. not very clean, not very noisy. It leads to decision problem: when to use *Mean Reversion* and when to use *Median Reversion.* This decision problem is tackled by *momentum* characteristic. If high momentum is associated with *Mean Reversion*, then *Mean Reversion* plays significant role in predicting price information. If high momentum is associated with *Median Reversion*, then *Median Reversion* plays significant role in predicting price information. Through this way, the proposed method intelligently tackles decision problem of *Mean* Vs. *Median* and combines both characteristics.

4.1.2 General Framework for Estimating Price Relatives

First step in portfolio investment requires estimation of price information of portfolio assets on the next trading period $(t+1)^{th}$ as accurate as possible. With this information, the investor would be able to put more capital on profitable assets. Several methods discussed in Section 3 involve prediction of price information too, followed by optimization step. Two recent methods *On-line Moving Average Mean Reversion* (OLMAR) [62] and *Robust Median Reversion* (RMR) [20] have shown that stock prices possess multi-period reversion property. This reversion property is exploited by two different approaches: 1) Mean Reversion [62] and 2) Median Reversion [20]. Empirically these two approaches show promising results. Median Reversion approach is more robust because it is not affected by noises and outliers severely unlike mean reversion approach.

This experiment aims to develop generalized approach for estimating price information for next trading period which exploits mean reversion and median reversion at the same time. The experiment proposes Bayesian Ensembling approach which effectively exploits reversion property from financial data by combining both *OLMAR* [62] and *RMR* [20] to make more accurate prediction of price-relatives for next trading period. Empirically, the algorithm using this approach outperforms current state-of-art algorithms on several datasets as indicated in Section 5.3.

The Bayesian Ensembling approach for on-line portfolio selection involves sequential Bayesian updating. The general framework for prediction using *BMMR* is given as follows:

$\tilde{x}_{t+1} = W_{t+1}^{OLMAR} * \tilde{x}_{t+1}^{OLMAR} + W_{t+1}^{RMR} * \tilde{x}_{t+1}^{RMR}$
such that $W_{t+1}^{OLMAR} + W_{t+1}^{RMR} = 1$

where, $\tilde{x}_{t+1}$ is the predicted price-relative on $(t+1)^{th}$ trading period, W_{t+1}^{OLMAR} is Bayesian weight of *OLMAR* on $(t+1)^{th}$ trading period and is also quantification of momentum associated with *Mean Reversion* characteristic, W_{t+1}^{RMR} is Bayesian weight of *RMR* on $(t+1)^{th}$ trading period and is also quantification of momentum associated with *Median Reversion* characteristic, $\tilde{x}_{t+1}^{OLMAR}$ is predicted price-relative on $(t+1)^{th}$ trading period using *OLMAR* approach and $\tilde{x}_{t+1}^{RMR}$ is predicted price-relative on $(t+1)^{th}$ trading period using *RMR* approach.

4.1.3 Sequential Bayesian Updating

Sequential Bayesian Updating provides a way to combine two different methods. It does so by calculating weights (or associated momentum) to be allocated to each individual prediction. Bayesian weights for each approach is calculated as:

$$W_{t+1}^{OLMAR} = W_{t}^{OLMAR} * L_{t+1}^{OLMAR};$$

Where, W_{t+1}^{OLMAR} is weight (or posterior probability) for *OLMAR* approach on the $(t+1)^{th}$ trading period, W_{t}^{OLMAR} is weight (or prior probability) on the t^{th} trading period and L_{t+1}^{OLMAR} is the likelihood function for *OLMAR* approach on the $(t+1)^{th}$ trading period. As mentioned in last subsection, this weight (or posterior probability) is quantification of momentum associated with *OLMAR* approach.

According to Officer [50], stock returns tend to follow normal distribution which is an exponential function. So, Likelihood function for *OLMAR* on any $(t+1)^{th}$ trading period is given as:

$$L_{t+1}^{OLMAR} = \frac{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{OLMAR}-x_{t,i})^2}}{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{OLMAR}-x_{t,i})^2}+e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{RMR}-x_{t,i})^2}}$$

where, $\tilde{x}_{t,i}^{OLMAR}$ is predicted price-relative for asset i for t^{th} trading period by the OLMAR prediction approach, $\tilde{x}_{t,i}^{RMR}$ is predicted price-relative for asset i for t^{th} trading period by the RMR prediction approach, $x_{t,i}$ is the actual price-relative for asset i for t^{th} trading period as revealed by the market and m is total number od assets in the portfolio.

Similarly, likelihood function and weight function for *RMR* can be also calculated.

4.2 Portfolio Optimization

Next step requires deciding proportion of the wealth being allocated to each asset of the stock based on the expected price-relatives of each asset received from the previous step. The experiment has adopted the idea of Passive Aggressive (PA) On-line Learning [51] to exploit mean reversion and median reversion for getting the maximized final cumulative wealth. It is similar to the one used in PAMR [21], OLMAR [19] and RMR [20]. It is formulated as follows:

$$b_{t+1} = \underset{b \in \Delta_m}{\text{argmin}} \frac{1}{2} \|b - b_t\|^2 \text{ such that } b \cdot \tilde{x}_{t+1} \geq \epsilon \qquad (2.0)$$

This equation aims at finding the optimal portfolio by minimizing the deviation from difference from the last portfolio, although it needs to satisfy one condition, i.e., $b \cdot \hat{x}_{t+1} \geq \epsilon$. The formulation effectively exploits reversion in price-relative information.

If the constraint of this formulation is satisfied, that is, if the expected return is higher than the threshold, i.e. ϵ then the resulting portfolio on $(t+1)^{th}$ trading period is same as the previous portfolio on t^{th} trading period. If the constraint is not satisfied, then it tries to calculate a new portfolio for $(t+1)^{th}$ trading period such that the expected return is greater than the threshold ϵ, minimizing the distance of this new portfolio from the previous portfolio, the one on the t^{th} trading period. The algorithm *BMMR* explicitly involves the reversion idea as it requires determination of the price-relative $\hat{x}_{t+1}$ for next trading period $t + 1^{th}$. Following the work of [56, 51, 19], the solution of optimization problem (2) is given as follows:

The optimization problem in terms of Lagrangian multiplier is:
$$\mathscr{L}(\mathbf{b}, \lambda, \eta) = \frac{1}{2} \|b - b_t\|^2 + \lambda(\epsilon - b \cdot \tilde{x}_{t+1}) + \eta(b \cdot 1 - 1) \qquad (2.1)$$

where η and non negative λ are Lagrangian multipliers.

In order to minimize the loss function $\mathscr{L}$, it's derivation with respect to $\mathbf{b}$ is taken and set to zero,
$$\frac{\partial \mathscr{L}}{\partial b} = (b - b_t) - \lambda(\epsilon - \tilde{x}_{t+1}) + \eta 1 = 0$$
$$\implies b = b_t + \lambda \tilde{x}_{t+1} - \eta 1 \qquad (2.2)$$

Multiplying both sides by 1^T leads to:
$$\implies 1 = 1 + \lambda \tilde{x}_{t+1} - \eta m$$
$$\implies \eta = \frac{\lambda \tilde{x}_{t+1}}{m}$$
$$\implies \eta = \lambda \bar{x}_{t+1} \qquad (2.3)$$

$\bar{x}_{t+1}$ (or $\frac{\tilde{x}_{t+1}}{m}$) is mean of expected price-relative which can be considered as market price. Using equations (2.1) and (2.2), it leads to:
$$b = b_t + \lambda(\tilde{x}_{t+1} - \bar{x}_{t+1} 1) \qquad (2.4)$$

Now, to get the Lagrangian multiplier:
$$\implies b - b_t = \lambda(\tilde{x}_{t+1} - \bar{x}_{t+1} 1) \qquad (2.5)$$

Using equations (2.1) and (2.5), it leads to:
$\implies \mathscr{L} = -\frac{1}{2}\lambda^2\|\tilde{x}_{t+1} - \bar{x}_{t+1}\|^2 + \lambda(\epsilon - b_t \cdot \tilde{x}_{t+1})$ (2.6)

Differentiating equation (2.6) with respect to λ and setting it to zero, leads to:
$\implies \frac{\partial \mathscr{L}}{\partial \lambda} = (\epsilon - b_t \cdot \tilde{x}_{t+1}) - \lambda\|\tilde{x}_{t+1} - \bar{x}_{t+1}1\|^2$
$\implies \lambda = \frac{\epsilon - b_t \cdot \tilde{x}_{t+1}}{\|\tilde{x}_{t+1} - \bar{x}_{t+1}1\|^2}$

So, the value of λ is given by:
$\lambda = \max 0, \frac{\epsilon - b_t \cdot \tilde{x}_{t+1}}{\|\tilde{x}_{t+1} - \bar{x}_{t+1}1\|^2}$

Here, the optimization solution of (2.0) without considering non-negativity constraint is given as follows:
$b_{t+1} = b_t + \lambda_{t+1}(\tilde{x}_{t+1} - \bar{x}_{t+1} \cdot 1)$
where $\bar{x}_{t+1} = \frac{1 \cdot \tilde{x}_{t+1}}{m}$ and $\lambda_{t+1} = \max\{0, \frac{\epsilon - b_t \cdot \tilde{x}_{t+1}}{\|\tilde{x}_{t+1} - \bar{x}_{t+1}1\|^2}\}$

Finally, to ensure that the portfolio b_{t+1} is non-negative, it is projected to the simplex domain [63].

4.3 Algorithm

First step of the model involves calculation of expected relative price on the next trading period. As mentioned in chapter 4.1, under *RMR* approach, L_1 median is determined as expected price for next trading and price-relative for the next period is determined as ratio to L_1 price to price on $(t-1)^{th}$ trading period [20]. Algorithm 2 shows steps to determine L_1 median. It uses the Modified Weiszfeld Algorithm [53] for it.

ALGORITHM 2: L_1 Median: Determining expected price-relative on next trading period

---

Input: Price p_t, p_{t-1}, ..., p_{t-w+1}; Maximum number of iterations m; Tolerance Level λ ;

Output: Expected price-relative $\tilde{x}_{t+1}$

Initialization $j = 2$; $\mu_1 = median(p_t, p_{t-1}, ..., p_{t-w+1})$;

Procedure:

while $j \leq m$ **do**

$$\eta(\mu) = \begin{cases} 1, & \text{if } \mu = p_{t-j} \\ 0, & \text{otherwise} \end{cases}$$

$\tilde{R}(\mu) = \sum_{p_{t-j} \neq \mu} \frac{p_{t-j} - \mu}{\| p_{t-j} - \mu \|}$ $\gamma(\mu) = \| \tilde{R}(\mu) \|$

$\tilde{\mu} = \frac{\sum_{p_{t-j} \neq \mu} \frac{p_{t-j}}{\| p_{t-j} - \mu \|}}{\sum_{p_{t-j} \neq \mu} \frac{1}{\| p_{t-j} - \mu \|}}$

$\mu = min(1, \frac{\eta(\mu)}{\gamma(\mu)}) + \tilde{\mu}(1 - \frac{\eta(\mu)}{\gamma(\mu)})^{+}$

if $(\| \mu_{j-1} - \mu_j \| \leq \lambda \| \mu_j \|)$ **then**

break

end

end

$\tilde{x}_{t+1} = \frac{\mu}{p_t}$

In Algorithm 2, $\|.\|$ represents Euclidean norm.

Algorithm 3 shows steps to estimate price-relatives of portfolio assets on the $(t+1)^{th}$ trading period based on *OLMAR* [62]. This algorithm is responsible for exploiting *Mean Reversion* characteristics in financial market.

ALGORITHM 3: OLMAR: Determining expected price-relative on next trading period

---

Input: window size w, Historical price-relative information upto t^{th} trading period: x_1^{t-1}

Output: Expected price-relative $\tilde{x}_{t+1}$

Procedure:

$\tilde{x}_{t+1} = \frac{MovingAverage_t(w)}{p_t} = \frac{1}{w}(1 + \frac{1}{x_t} + ... + \frac{1}{\odot_{i=0}^{w-2} x_{t-i}})$

Algorithm 4 shows steps to determine final price-relatives though Bayesian ensembling of *Mean Reversion* and *Median Reversion* characteristics of financial market as given by Algorithm 2 and Algorithm 3 respectively.

ALGORITHM 4: Bayesian Ensembling: Ensembling price-relatives predicted by Algorithm 2 and Algorithm 3

Input: Historical Price Information: p_1^t, Historical Price-Relative Information: x_1^t, Maximum Number of iterations: m, Tolerance Level: λ

Output: Expected price-relative $\tilde{x}_{t+1}$ on the $(t+1)^{th}$ trading period

Initialization $W_1^{OLMAR} = W_1^{RMR} = 0.5$, $\mu_1 = median(p_t, p_{t-1}, ..., p_{t-w+1})$;

Procedure:

Expected Price-Relative given by Algorithm 2

$$\tilde{x}_{t+1}^{RMR} = \text{Algorithm 2: } L_1 Median(p_1^t, m, \lambda)$$

Expected Price-Relative given by Algorithm 3

$$\tilde{x}_{t+1}^{OLMAR} = \text{Algorithm 3: } OLMAR(x_1^t, w)$$

Calculate final expected price on $(t+1)^{th}$ trading period using Bayesian ensembling;

$$\tilde{x}_{t+1} = W_{t+1}^{RMR} * \tilde{x}_{t+1}^{RMR} + W_{t+1}^{OLMAR} * \tilde{x}_{t+1}^{OLMAR}$$

Update weights;

$$L_{t+1}^{RMR} = \frac{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{RMR} - x_{t,i})^2}}{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{RMR} - x_{t,i})^2} + e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{RMR} - x_{t,i})^2}}$$

$$L_{t+1}^{OLMAR} = \frac{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{OLMAR} - x_{t,i})^2}}{e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{OLMAR} - x_{t,i})^2} + e^{-\sum_{i=1}^{m}(\tilde{x}_{t,i}^{RMR} - x_{t,i})^2}}$$

$$W_{t+1}^{RMR} = W_t^{RMR} * L_{t+1}^{RMR}$$

$$W_{t+1}^{OLMAR} = W_t^{OLMAR} * L_{t+1}^{OLMAR}$$

Algorithm 5 shows steps to determine investment strategy on the $(t+1)^{th}$ trading period using price-relative information predicted by Algorithm 4, Algorithm 3 and Algorithm 2. It is responsible for deciding how much capital should be invested any particular asset of the portfolio on the $(t+1)^{th}$ trading period. It uses on-line passive aggressive machine learning algorithm to decide allocation [51, 62].

ALGORITHM 5: BMMR: Determining allocation of the wealth to be invested in each portfolio asset on the $(t+1)^{th}$ trading period

Input: Threshold Reversion ϵ, Predicted next price-relative $\tilde{x}_{t+1}$, Current Portfolio b_t

Output: Next Portfolio b_{t+1}

Procedure:

Calculate the predicted Market Return;

$$\bar{x}_{t+1} = \frac{1^T \tilde{x}_{t+1}}{m}$$

Calculate hinge loss function;

$$loss = \epsilon - b_t \cdot \tilde{x}_{t+1}$$

Calculate the Lagrangian multiplier;

$$\lambda_{t+1} = \max\{0, \frac{loss}{\|\tilde{x}_{t+1} - \bar{x}_{t+1} \cdot 1\|^2}\}$$

Update the portfolio;

$$b_{t+1} = b_t + \lambda_{t+1}(\tilde{x}_{t+1} - \bar{x}_{t+1} \cdot 1)$$

Normalize the portfolio b_{t+1};

$$b_{t+1} = \underset{b \in \Delta_m}{\operatorname{argmin}} \|b - b_{t+1}\|^2$$

Algorithm 6 shows overall framework to determine investment strategy on the $(t+1)^{th}$ trading period which exploits both *Mean Reversion* and *median Reversion* characteristics of the financial market & combine it using Bayesian Emsembling approach.

ALGORITHM 6: On-line Portfolio Selection using BMMR

Input: Window w ≥ 2; price-relative x_1^n: From 1^{st} trading period to t^{th} trading period; Maximum number of iterations m; Tolerance Level λ; Threshold Reversion ϵ; Initial Wealth C_0; Initial Investment Strategy $b_0 = \{ \frac{1}{m}, ..., \frac{1}{m} \}$

Output: C_n : Final cumulative wealth after n^{th} trading period

Initialization $i = 1$; $t = 1$; Price Information $p_0 = 1$;

Procedure:

Convert price-relative information into actual price information on any period t

while $t \leq n$ **do**

Extract price information from price-relative information $p_t = p_{t-1} \cdot x_i$

Investor formulates portfolio b_t based on historical price information $(b^{t-1}, x^{t-1}, p^{t-1})$

Actual price-relative on trading period t is generated by the market i.e. x_t

Portfolio daily return is given by $b_t^T x_t$ and the cumulative wealth is updated to $C_t = C_{t-1} \times b_t^T x_t$

Investor predicts price-relatives of portfolio assets for the next trading period t+1

$\tilde{x}_{t+1}$= Algorithm 4: Bayesian Ensembling(p_1^t, x_1^t, m, λ)

Investor updates its portfolio selection strategy according to:

b_{t+1}= Algorithm 5: BMMR(ϵ, $\tilde{x}_{t+1}$, b_t)

t = t + 1

end

4.4 Computational Analysis of the Algorithm

Time complexity of trading algorithms play significant role in high frequency trading environment where thousands of transactions take place in fraction of seconds [66]. Time complexity of different algorithms have been reported in Table 4.1.

Trading Algorithm	Time Complexity
UP	$O(n^m)/O(m^7n^8)$
EG	$O(mn)$
ONS	$O(m^3n)$
B^K	$O(e^2mn^2) + O(emn^2)$
B^{NN}	$O(e^2mn^2) + O(emn^2)$
CORN	$O(e^2mn^2) + O(emn^2)$
Anticor	$O(e^3m^2n)$
PAMR	$O(mn)$
CWMR	$O(mn)$
OLMAR	$O(mn)$
RMR	$O(mn) + O(ln)$
BMMR	$O(mn) + O(ln)$

Table 4.1: Time complexity of different trading algorithms. m represents the number of stocks; n represents the number of trading periods; e represents the number of experts; $\boldsymbol{l}$ represents the maximum number of iterations in Algorithm 1.

It can observed that BMMR is linear with respect to number of stocks in the portfolio $\boldsymbol{m}$ and number of trading periods $\boldsymbol{n}$. Algorithm 2 is implemented in *O(l)* as it involves maximum number of $\boldsymbol{l}$ iterations. So, Algorithm 2 takes *O(l)* per period. Algorithm 3 takes *O(m)* per period. In addition to that, Algorithm 4 takes *O(m)* per period. So, the total time complexity becomes *O(mn)* + *O(mn)*+ *O(ln)*, i.e. *2O(mn)*+ *O(ln)*. Neglecting coefficients, the eventual time complexity of BMMR as a whole algorithm is *O(mn)*+ *O(ln)*.

Chapter 5. Experiments

5.1 Datasets

In the experiment, 6 different datasets have been used. Each dataset contains historical relative daily prices of stocks. Each dataset is publicly available. Moreover, it can extracted from public domain like Google Finance and Yahoo Finance etc. All these 6 datasets represent several financial markets.

1. **NYSE (O)**: It is one of the oldest and standard benchmark datasets for testing any kind of financial optimization problem involving stock prices. It is named as NYSE (Old) or simply NYSE (O). It was first used by [29] and later on, by [34], [19], [31], [31], [33], [8] and [11] to devise various types of investment strategies. This dataset contains 5651 daily price-relatives of 36 stocks listed on New York Stock Exchange Market for a period of 22 years starting from 3 July 1962 to 31 December 1984.

2. **NYSE (N)**: This dataset is extended version of previous NYSE dataset. It is called NYSE (New) or simply NYSE (N). It contains historical daily price-relatives of 23 stocks listed on New York Exchange Market for period of 6431 trading periods starting from 1 January 1985 to 30 June 2010. It contains fewer stocks than NYSE (O) contains because some of companies were taken over by other companies or went bankrupt.

3. **TSE**: This dataset was collected by Borodin et. al. [31]. It contains daily price-relatives of 88 stocks from Toronto Stock Exchange for a period of 1259 trading periods starting from 4 January 1994 to 31 December 1998.

4. **SP500**: This dataset was also collected by Borodin et. al. [31]. It contains daily price-relatives of 25 companies which have the largest market capitalization among 500 SP500 companies for a period of 1276 trading periods starting from 2 January 1998 to 31st January 2003.

5. **MSCI**: This dataset consists of global equity indices. These indices are used to make the MSCI World Index. This dataset of collection of 24 indices which represent the equity markets of 24 countries around the globe. The dataset contains daily price-relatives for a period of 1043 trading periods starting from 1 April 2006 to 31 March 2010. It is maintained by MSCI Inc., previously known as Morgan Stanly Capital International.

6. **DJIA**: This dataset is collection of 30 Dow Jones composite stocks. It contains daily price-relatives for a period of 507 trading periods starting from 14 January 2001 to 14 January 2003.

In Table 5.1, all these 6 datasets have been summarized.

Dataset	Market	Region	Time Frame	#Trading periods	#Assets
NYSE (O)	Stock	USA	3rd July 1962 - 31st December 1994	5651	36
NYSE (N)	Stock	USA	1st January 1985 - 30th June 2010	6431	23
TSE	Stock	Canada	4th January 1994 - 31st December 1998	1259	88
SP500	Stock	USA	2nd January 1998 - 31st January 2003	1276	25
MSCI	Index	Global	1st April 2006 - 31st March 2010	1043	24
DJIA	Stock	USA	14th January 2001 - 14th January 2003	507	30

Table 5.1: Summary of all six datasets from real financial markets.

5.2 Experimental Setup

5.2.1 Metrics

In this section, we describe various metrics which we used to measure then performance of our algorithm.

1. **Cumulative Wealth**: Cumulative wealth is the aggregate amount that the given investment has generated over a fixed period of time. Higher is its value, better is the investment strategy [64].

2. **Standard Deviation**: Standard Deviation is a measure the dispersion of a set of data from its mean. In finance, it is used to measure volatility of given investment strategy. It is widely used as a metric to estimate amount of expected volatility. This metric is quite important for risk averse investors. Lower is its value, better is the investment strategy [64].

3. **Sharpe Ratio**: Sharpe Ratio is a measure for calculating risk-adjusted return, and this ratio has become the industry standard for such calculations. The Sharpe ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. The performance associated with risk-taking activities is determined by subtracting the risk-free rate from the mean return. Higher is the value of the Sharpe ratio, better is the investment strategy.[64].

4. **Calmar Ratio**: It is comparison of the average annual compounded rate of return and the maximum drawdown risk of commodity trading advisors and hedge funds. Lower is its value, worse is the performance of investment on a risk-adjusted basis over the specified time period; the higher the Calmar Ratio, the better it performed. In general, the time period used is three years, but this can be higher or lower based on the investment under consideration [64].

5. **Sortino Ratio**: A modification of the Sharpe ratio that differentiates harmful volatility from general volatility by taking into account the standard deviation of negative asset returns, called downside deviation. The Sortino ratio subtracts the risk-free rate of return from the portfolio's return, and then divides that by the downside deviation. A large Sortino ratio indicates there is a low probability of a large loss. So, higher is the value of the Sortino ratio, better is the investment strategy. It is calculated as follows [64]:
   Sortino Ratio = $\frac{<R> - R_f}{\sigma_d}$
   Where $< R >$ is the Expected Return, R_f is the Risk free Rate of Return and σ_d is Standard Deviation of Negative Asset Returns.

6. **Maximum Drawdown**: A maximum drawdown (MDD) is the maximum loss from a peak to a trough of a portfolio, before a new peak is attained. Maximum Drawdown (MDD) is an indicator of downside risk over a specified time period. It can be used both as a stand-alone measure or as an input into other metrics such as "Return over Maximum Drawdown" and Calmar Ratio. Lower is its value, better is the investment strategy. Maximum Drawdown is expressed in percentage terms and computed as [64]:

$$(\text{Trough Value} - \text{Peak Value})/\text{Peak Value}$$

The performance of the proposed algorithm has been measured and presented using all six metrics, but the primary focus is on Cumulative Wealth. The experiment is specifically focused on maximizing its value as consequences of the algorithm.

5.2.2 Comparison Approaches

The proposed algorithm has been compared with a number of existing benchmark algorithms and other non benchmark algorithms having better empirical performance , introduced by Computer Science community. Here, a list of these algorithms have been provided with their parameters as introduced in their original studies. Those algorithms which are focused on theoretical analysis only in this field, have not been considered.

1. **Market**: Simply once buy-then-hold-uniformly and no parameter.
2. **Best Stock**: Follow the best performing stock in the hindsight. Knowledge of the best stock is known in hindsight. No parameter.
3. **BCRP**: Follow the Best Constant Rebalanced Portfolio strategy in hindsight. No parameter.
4. **UP**: This algorithm was introduced by Cover at. al. [29]. We focus on its implementation by Kalai et. al. It's parameters are: $\delta_0 = 0.004$, $\delta = 0.005$, $m = 100$ and $S = 500$ [65].
5. **EG**: Exponential Gradient Algorithm with learning rate parameter: $\eta = 0.05$ [33].
6. **ONS**: on-line Newton Step introduced with parameters: $\eta = 0$, $\beta = 1$, $\gamma = 0.125$ [34].
7. **Anticor**: Exploiting *Mean Reversion* characteristics using cross-correlation and auto-correlation. No parameter. [31].
8. $\mathbf{B}^K$: Nonparametric Kernel based moving window strategy with parameters: $w = 5$, $L = 10$ and correlation coefficient threshold $\epsilon = 0.1$ [8].
9. $\mathbf{B}^{NN}$: Non Parametric Nearest Neighbour based Strategy with parameters: $w = 5$, $L = 10$ and $p_l = 0.02 + 0.5(l-1)/(L-1)$ [11].
10. **CORN**: Correlation Driven Non-Parametric based Strategy with parameters: $w = 5$, $p = 1$ and $\rho = 0.1$ [13].
11. **PAMR**: Passive Aggressive Mean Reversion with parameters: $\epsilon = 0.5$ [21].
12. **CWMR**:Confidence Weighted Mean Reversion with parameter: $\epsilon = 0.5$ [57].
13. **OLMAR**: on-line Moving Average Reversion with parameters: $\epsilon = 10$ and $w = 5$ [19].
14. **RMR**: Robust Median Reversion with parameters: $\epsilon = 10$ and $w = 5$ [20].

5.3 Experimental Results

5.3.1 Cumulative Wealth

Method	NYSE (O)	NYSE (N)	TSE	SP500	MSCI	DJIA
Market	14.50	18.06	1.61	1.34	0.91	0.76
Best - Stock	54.14	83.51	6.28	3.78	1.50	1.19
BCRP	250.60	120.32	6.78	4.07	1.51	1.24
UP	26.68	31.49	1.60	1.62	0.92	0.81
EG	27.09	31.00	1.59	1.63	0.93	0.81
ONS	109.19	21.59	1.62	3.34	0.86	1.53
B^K	1.08E+09	4.64E+03	1.62	2.24	2.64	0.68
B^{NN}	3.35E+11	6.80E+04	2.27	3.07	13.47	0.88
CORN	1.48E+13	5.37E+05	3.56	6.35	**26.10**	0.84
Anticor	2.41E+08	6.21E+06	39.36	5.89	3.22	2.29
PAMR	5.14E+15	1.25E+06	264.86	5.09	15.23	0.68
CWMR	6.49E+15	1.41E+06	332.62	5.90	17.28	0.68
OLMAR	4.04E+16	2.24E+08	**424.80**	5.83	16.33	2.12
RMR	**1.64E+17**	3.25E+08	181.34	8.28	16.76	2.67
BMMR	9.43E+16	**4.74E+08**	60.97	**12.79**	14.02	**2.77**

Table 5.2: Cumulative Wealth achieved by various strategies on the six datasets. The best results in each dataset are highlighted in **bold**.

In Table 5.2, Cumulative wealth achieved by the algorithm (BMMR) has been reported. From the table, it can be observed that BMMR outperforms all state-of-the-art methods and benchmarks methods on several datasets: NYSE(N), SP500 and DJIA. On other datasets NYSE(O), TSE and MSCI, it outperforms all benchmark algorithms and most of non benchmark algorithms.

5.3.2 Sharpe Ratio

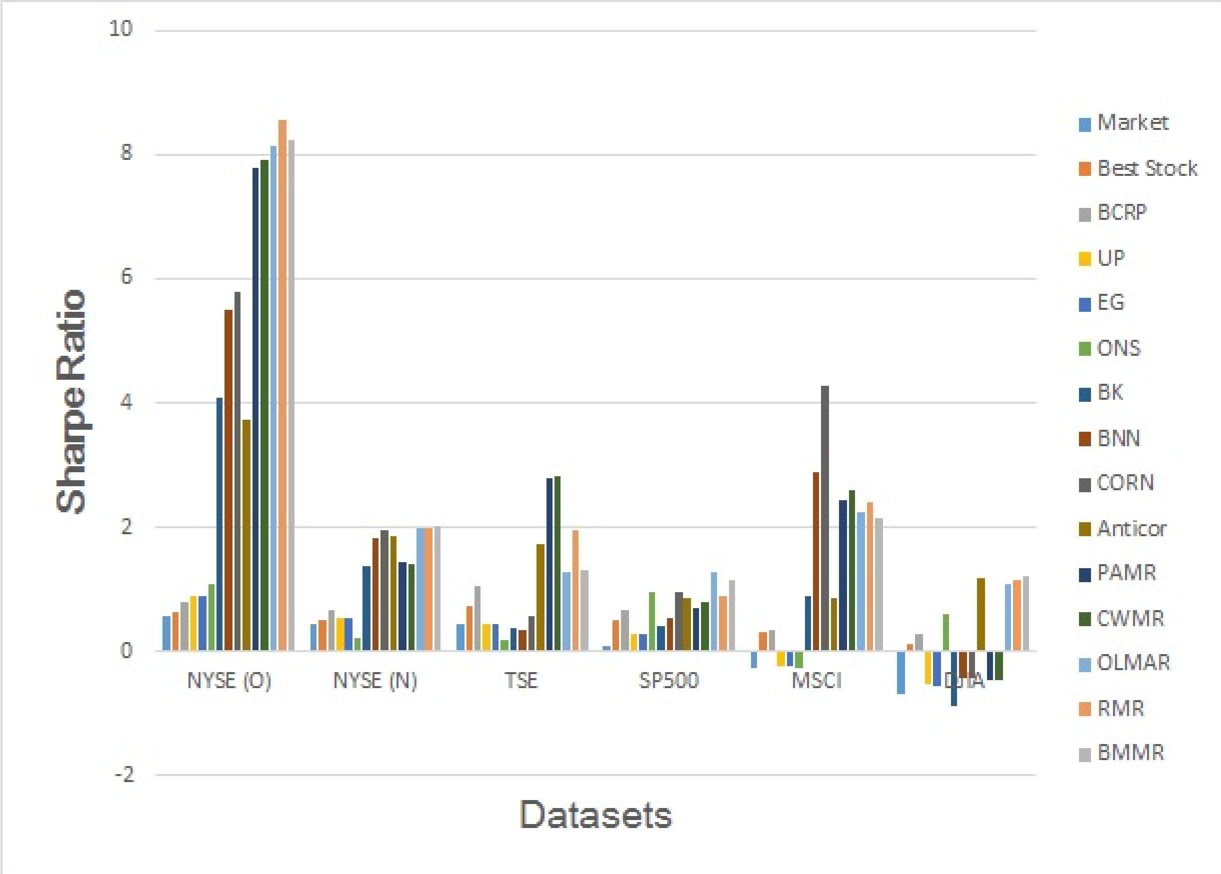

Figure 5.1: Sharpe Ratio

Figure 5.1 shows the Sharpe ratio achieved by BMMR and benchmarks as wll as other state-of-the-art methods on all six different datasets. Higher Sharpe ratio is always preferred. From the Figure 5.1, it can be observed that the Sharpe ratio of BMMR is more or less same as achieved by other methods. So, it can be concluded that for a given amount of risk, returns produced by BMMR is high which makes it a safe investment strategy.

5.3.3 Standard Deviation

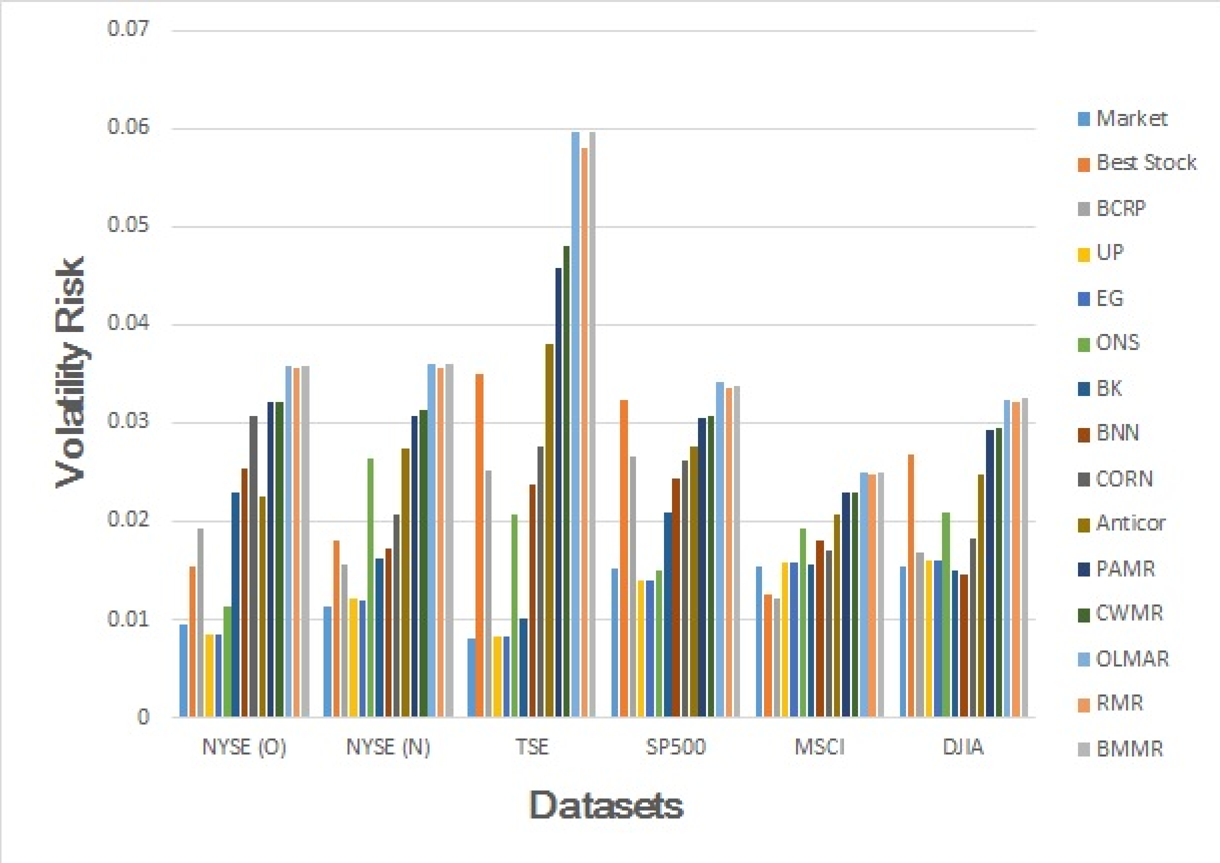

Figure 5.2: Volatility Risk

Figure 5.2 shows the standard deviation achieved by the BMMR and benchmarks as well as other state-of-the-art methods on all six datasets. Standard deviation is considered as a metric to measure volatility risk. In financial market, higher return always come at the cost of higher risk. Even after that, volatility risk of BMMR is comparable to the other algorithms which also achieved nearly same or less return. So, in light of the cumulative wealth achieved by the BMMR, volatility risk demonstrated it, is justified.

5.3.4 Calmar Ratio

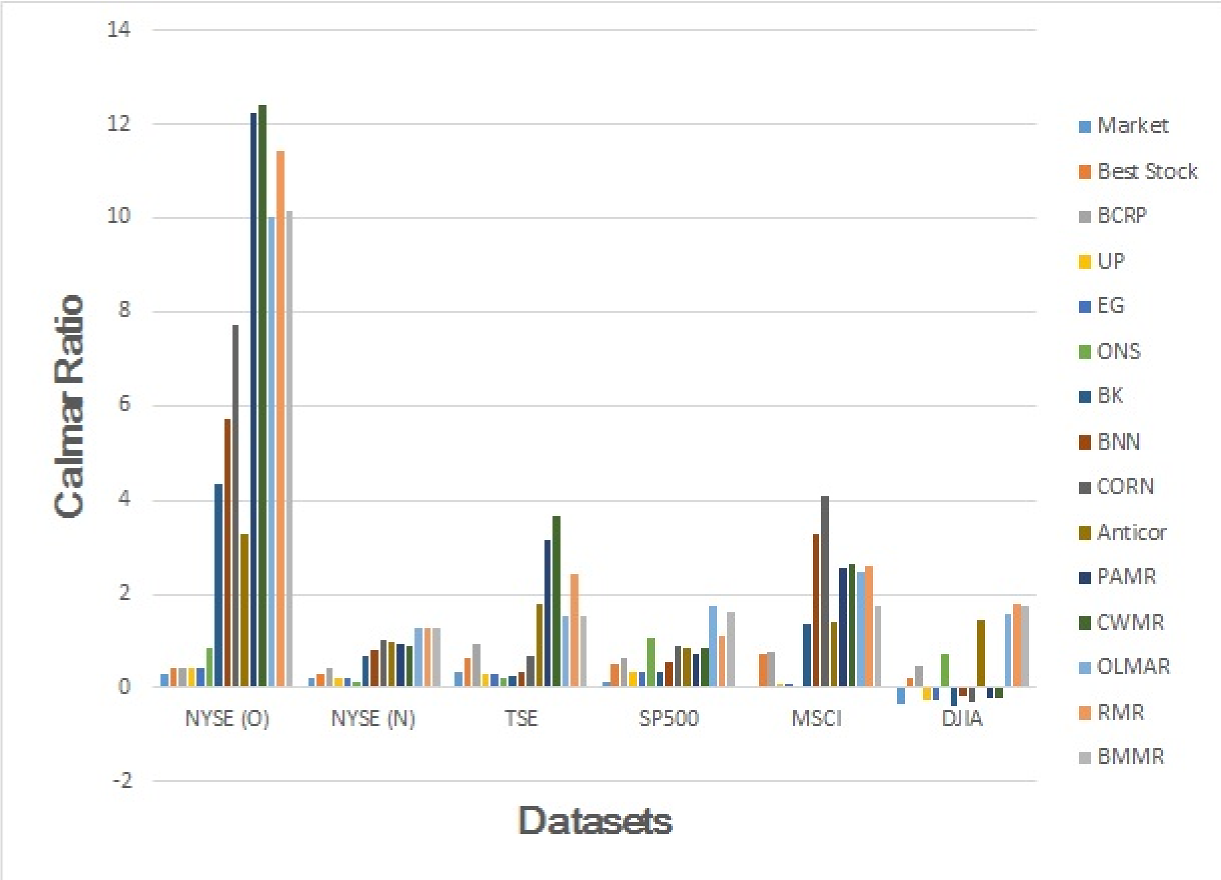

Figure 5.3: Calmar Ratio

Figure 5.3 shows the Calmar ratio achieved by the BMMR and benchmarks as well as other state-of-the-art methods on all six datasets. In can observed that Calmar ratio of BMMR is not drastically low. It's more or less same as other state-of-the-art algorithms. Generally higher Calmar ratio is preferred. So, it can be concluded that BMMR performs well or almost same as other state-of-the-art methods on a risk-adjusted basis over the specified time period.

5.3.5 Sortino Ratio

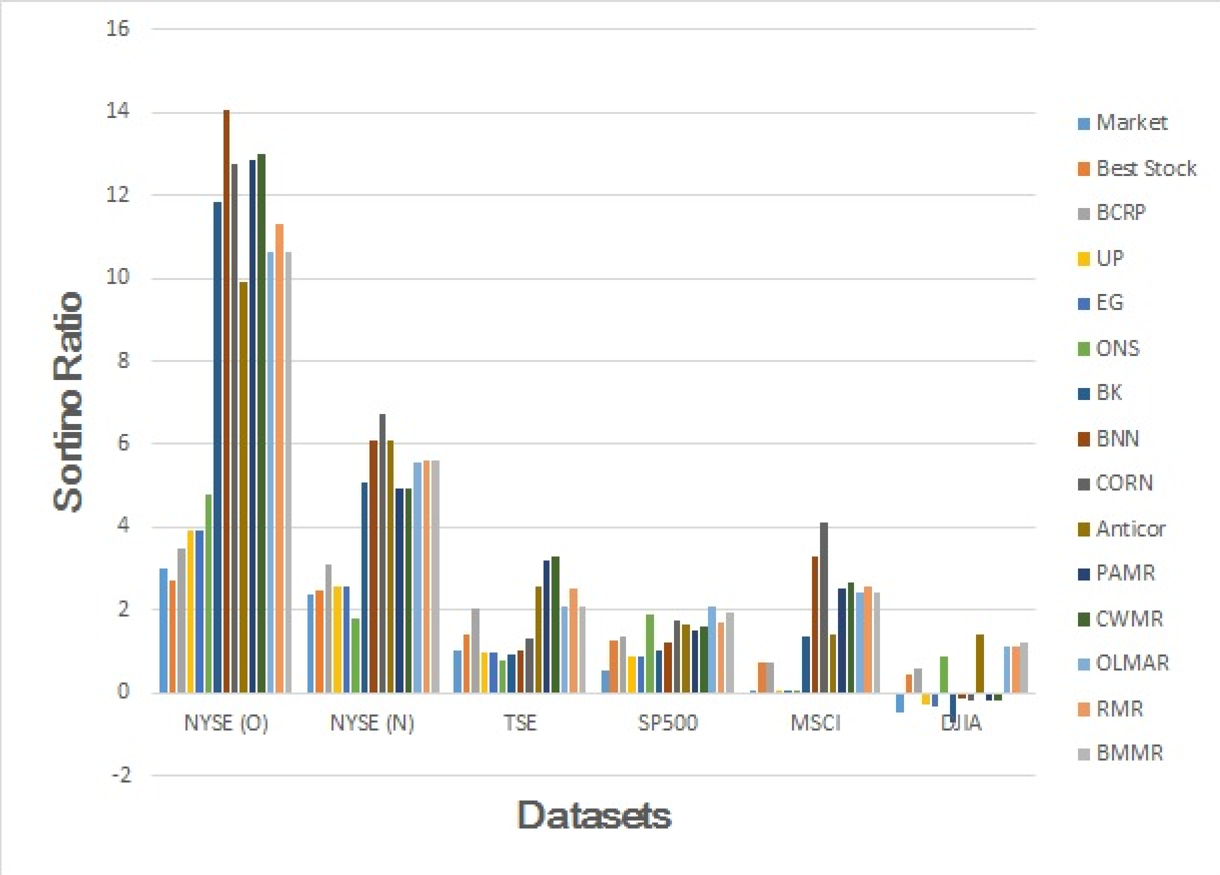

Figure 5.4: Sortino Ratio

Figure 5.4 shows the Sortino ratio achieved by BMMR and other methods on all datasets. The Sortino ratio achieved by BMMR is quite high. It indicates that chance of suffering from the large loss is quite low while following this BMMR based strategy.It confirms that that BMMR is relatively stable in terms of its wealth generation capacity.

5.3.6 Maximum Drawdown

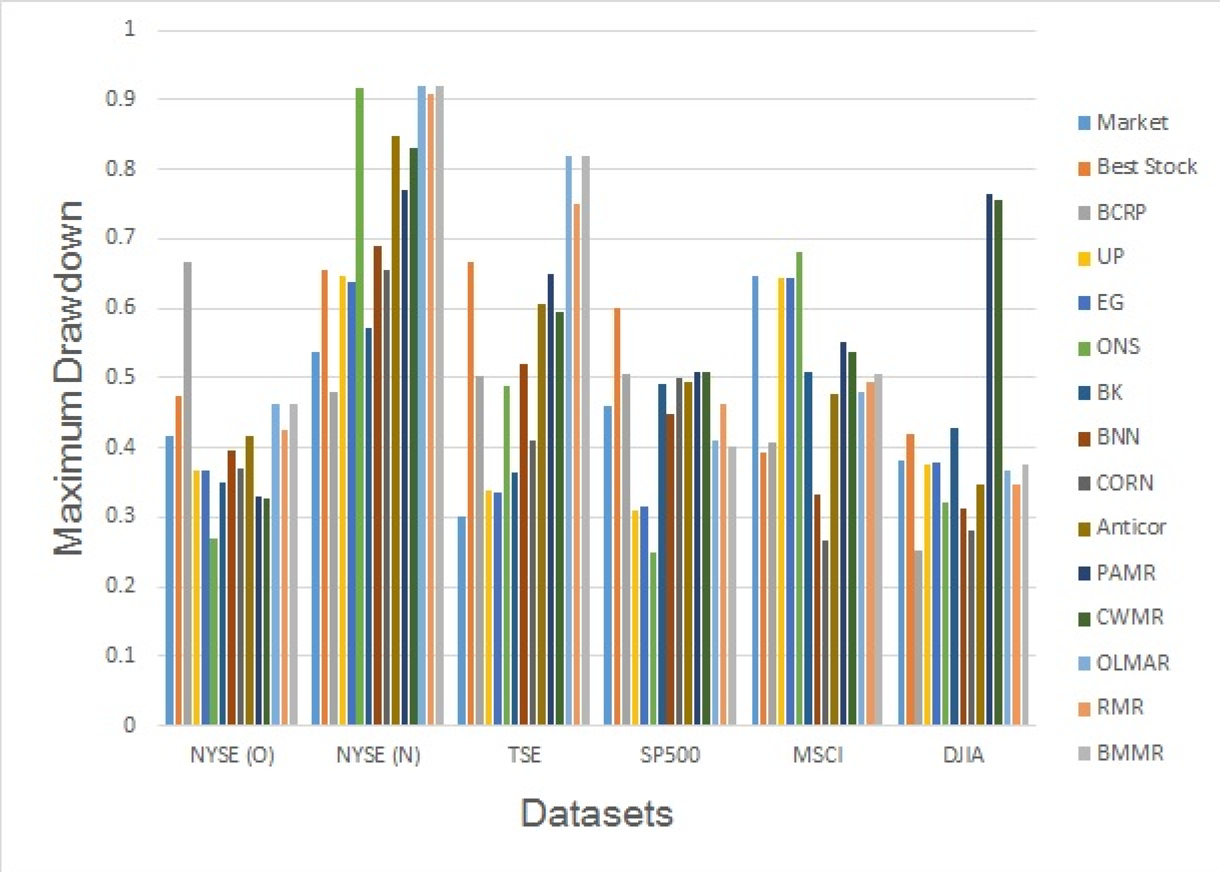

Figure 5.5: Maximum Drawdown

Figure 5.5 shows performance of BMMR and other benchmark as well as state-of-the-art methods based on *Maximum Drawdown* metric. From the figure, it can be observed from results on several datasets that Maximum drawdown of BMMR is not significantly high in light of its wealth. It's nearly same as demonstrated by other state-of-the-art-algorithms. Chances of BMMR getting declined from its historical peak is relatively low. So, BMMR is not highly risky strategy.

5.3.7 Computational Cost

In Table 5.3, time taken by each algorithm has been reported on all 6 datasets. Time has been measured in seconds. All algorithms have been tested on a computer with capacity of 16 GB RAM and Ubuntu 14.04 as its operating system.

Algorithm	NYSE (O)	NYSE (N)	TSE	SP500	MSCI	DJIA
UP	582.17	510.58	74.83	26.87	14.04	6.27
EG	3.29	2.73	0.69	0.27	0.32	0.10
ONS	1108.44	847.03	401.23	24.37	17.88	12.42
B^K	33990.44	29072.16	6025.23	3070.35	2927.034	1358.39
B^{NN}	26596.58	22041.31	13608.67	2529.60	2566.47	1252.86
CORN	14461.36	16430.35	2198.38	791.49	608.58	197.36
Anticor	194.78	57.27	53.79	12.69	13.49	3.34
PAMR	1.18	1.41	0.30	0.23	0.18	0.08
CWMR	2.47	2.36	1.44	0.45	0.38	0.21
OLMAR	1.24	1.29	0.25	0.23	0.17	0.10
RMR	3.76	3.72	0.82	0.75	0.73	0.32
BMMR	4.57	4.98	1.05	0.83	0.68	0.34

Table 5.3: Computational time (in seconds) on the six real datasets.

As clearly shown in Table 5.3, in light total wealth generated by BMMR, time taken by BMMR is drastically low, this making it very suitable for large scale or high frequency financial applications where computation efficiency and final cumulative wealth are paramount issues.

5.3.8 Performance Analysis Dataset-Wise

As shown in Table 5.2, the proposed method achieves superior performance on 3 datasets. On other 3 datasets, other methods achieve superior performance. The performance on each dataset depends on characteristics embedded in it. Moreover, financial data are very complex in nature. They contain multiple characteristics. In this section, each dataset has been discussed individually to find conditions or characteristics under which the proposed algorithm achieves the superior performance.

1. **NYSE (O)**: This is the oldest benchmark dataset recorded from 1962 to 1994 from New York exchange market. During this time, computers were not used well to record data and to process transactions. Also, the hardware were not highly sophisticated as they are now. As a result, this dataset contains lots of noises and outliers. That's why algorithm using this portfolio (or dataset) must be robust to noises and outliers. *Median Reversion* offers the most ideal choice due to its robust nature. RMR is entirely based on *Median Reversion* and achieves the most superior performance. The proposed method sometimes uses *mean reversion* and sometimes uses *median reversion*, but not *median reversion* all the time like RMR.

2. **NYSE (N)**: This dataset was recorded between 1985 to 2010 from New York exchange market. During this time, computers were used to record data and to process transactions. Hardware used at exchange market were also relatively sophisticated. As a result, this dataset is relatively clean i.e. cleaner than NYSE (O). Empirical experiments have already shown that reversion characteristics is quite significant for this dataset. The proposed method uses both *mean* and *median* to exploit reversion characteristics and achieves the most superior performance.

3. **TSE**: This dataset was recorded between 1994 and 1998 from Toronto exchange market (Canada). During this time, Canadian stock market was stable unlike financial crisis in East Asian stock market. It didn't go through major ups-downs, noises or outliers. So it does not need to use robust metric like median. In absence of noises or outliers, reversion characteristics is best exploited by *mean*. So, *mean reversion* based OLMAR achieves the best performance.

4. **SP500**: This dataset recorded between 1998 and 2003, contains price information of 25 companies which have the largest market capitalization among all SP500 companies. These 25 companies include Internet based companies like Microsoft, Apple, Verizon, Amazon etc. which suffered from Dot-Com bubble during this period. Also, other big companies which were not based on Internet, suffered from East Asian financial crisis. Due to these crisis, stock prices of these 25 companies suffered from major ups-downs and outliers. But the dataset is still relatively clean due to sophisticated hardware used to store and process transactions. Empirical experiments have already shown that reversion characteristics is quite significant for this dataset. The proposed method uses both *mean* and *median* to exploit reversion characteristics and achieves the most superior performance.

5. **MSCI**: This dataset was recorded between 2006 and 2010. This period experienced the worst global financial crisis ever since 1929 financial crisis. During major global crisis, prices of all stocks of the portfolio become correlated. This characteristics has been also observed in past major financial crisis. This characteristics can be exploited via learning through history. *Reversion* and *Momentum* characteristics are not major ones during major crisis time. So, the method which learns from history, performs very well during crisis. CORN achieves the best performance on this

dataset as it uses correlation to learn from the historical prices. The proposed method does not exploit this characteristics resulting into relatively less performance that MSCI's.

6. **DJIA**: This dataset, recorded between 1998 and 2003, contains price information of 30 Dow Jones composite stocks. These stocks include companies like Microsoft, Intel AT & T Corporations etc which suffered from Dot-Com bubble. Due to this bubble, stock prices suffered from major ups-downs and outliers. But the dataset is still relatively clean due to sophisticated hardware used to store and process transactions. Empirical experiments have already shown that reversion characteristics is quite significant for this dataset. The proposed method uses both *mean* and *median* to exploit reversion characteristics and achieves the most superior performance.

After discussing these 6 datasets, it can be stated that the proposed method achieves the superior performance when the dataset contains reversion property and contains certain amount of noises/outliers. Financial data are complex in nature and almost always contain outliers & noises to some extent. The proposed method works well because it exploits *reversion* efficiently and also, it is robust to noises and outliers.

Chapter 6. Conclusion

In this paper, a novel multiple period on-line portfolio selection strategy named "Bayesian ensembled Mean-Median Reversion" (BMMR) has been proposed which exploits both *Reversion* characteristic & *Momentum* characteristic and tackles decision problem of *Mean Reversion* Vs. *Median Reversion* using Bayesian ensembling approach. The method also shows its effectiveness by achieving promising results on datasets extracted from many real world financial markets around the world. The algorithm is also efficient computationally which makes it quite suitable for large scale and high frequency trading applications.

Directions of future research include incorporation of transaction costs and taxes into the portfolio selection model. Another possible direction is to investigate different ensemble learning models like Bagging, Boosting and Stacking to tackle decision problem.

References

[1] Li, B. and Hoi, S.C., 2014. Online portfolio selection: A survey. ACM Computing Surveys (CSUR), 46(3), p.35.

[2] Breiman, L., 1961. Optimal gambling systems for favorable games. The Kelly Capital Growth Investment Criterion, p.47.

[3] Finkelstein, M. and Whitley, R., 1981. Optimal strategies for repeated games. Advances in Applied Probability, pp.415-428.

[4] Domowitz, I. and Yegerman, H., 2005. The cost of algorithmic trading: a first look at comparative performance. Trading, 2005(1), pp.30-40.

[5] Almgren, R. and Chriss, N., 2001. Optimal execution of portfolio transactions. Journal of Risk, 3, pp.5-40.

[6] Cover, T.M. and Gluss, D.H., 1986. Empirical Bayes stock market portfolios. Advances in applied mathematics, 7(2), pp.170-181.

[7] Kirkpatrick II, C.D. and Dahlquist, J., 2010. Technical analysis: the complete resource for financial market technicians. FT press.

[8] Györfi, L., Lugosi, G. and Udina, F., 2006. Nonparametric kernel-based sequential investment strategies. Mathematical Finance, 16(2), pp.337-357.

[9] Györfi, L. and Schäfer, D., 2003. Nonparametric prediction. Advances in learning theory: methods, models and applications, 339, p.354.

[10] Györfi, L., Ottucsák, G. and Walk, H. eds., 2012. Machine learning for financial engineering (Vol. 8). World Scientific.

[11] Györfi, L., Udina, F. and Walk, H., 2008. Nonparametric nearest neighbor based empirical portfolio selection strategies. Statistics & Decisions International mathematical journal for stochastic methods and models, 26(2), pp.145-157.

[12] Benesty, J., Chen, J., Huang, Y. and Cohen, I., 2009. Pearson correlation coefficient. In Noise reduction in speech processing (pp. 1-4). Springer Berlin Heidelberg.

[13] Li, B., Hoi, S.C. and Gopalkrishnan, V., 2011. Corn: Correlation-driven nonparametric learning approach for portfolio selection. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), p.21.

[14] Markowitz, H., 1952. Portfolio selection. The journal of finance, 7(1), pp.77-91.

[15] Kelly Jr, J.L., 1956. A new interpretation of information rate. Information Theory, IRE Transactions on, 2(3), pp.185-189.

[16] Browne, S. and Whitt, W., 1996. Portfolio choice and the Bayesian Kelly criterion. Advances in Applied Probability, pp.1145-1176.

[17] Györfi, L., Urbán, A. and Vajda, I., 2007. Kernel-based semi-log-optimal empirical portfolio selection strategies. International Journal of Theoretical and Applied Finance, 10(03), pp.505-516.

[18] Vajda, I., 2006. Analysis of semi-log-optimal investment strategies. In Prague Stochastics (pp. 719-727).

[19] Li, B., Hoi, S.C., Sahoo, D. and Liu, Z.Y., 2015. Moving average reversion strategy for on-line portfolio selection. Artificial Intelligence, 222, pp.104-123.

[20] D. Huang, J. Zhou, B. Li, S.C.H. Hoi, and S. Zhou. Robust median reversion strategy for on-line portfolio selection. In Proceedings of the 23th International Joint Conference on Artificial Intelligence, pages 2006–2012. AAAI Press, 2013.

[21] Li, B., Hoi, S.C., Zhao, P. and Gopalkrishnan, V., 2013. Confidence weighted mean reversion strategy for online portfolio selection. ACM Transactions on Knowledge Discovery from Data (TKDD), 7(1), p.4.

[22] Györfi, L. and Vajda, I., 2008, October. Growth optimal investment with transaction costs. In Algorithmic Learning Theory (pp. 108-122). Springer Berlin Heidelberg.

[23] Ottucsák, G. and Vajda, I., 2007. An asymptotic analysis of the mean-variance portfolio selection. Statistics & Decisions, 25(1/2007), pp.63-86.

[24] Cesa-Bianchi, N. and Lugosi, G., 2006. Prediction, learning, and games. Cambridge university press.

[25] Vovk, V.G., 1990. Aggregating strategies. In Proc. Third Workshop on Computational Learning Theory (pp. 371-383). Morgan Kaufmann.

[26] Vovk, V. and Watkins, C., 1998, July. Universal portfolio selection. In Proceedings of the eleventh annual conference on Computational learning theory (pp. 12-23). ACM.

[27] Akcoglu, K., Drineas, P. and Kao, M.Y., 2002. Fast universalization of investment strategies with provably good relative returns. In Automata, languages and programming (pp. 888-900). Springer Berlin Heidelberg.

[28] Akcoglu, K., Drineas, P. and Kao, M.Y., 2004. Fast universalization of investment strategies. SIAM Journal on Computing, 34(1), pp.1-22.

[29] Cover, T.M., 1991. Universal portfolios. Mathematical finance, 1(1), pp.1-29.

[30] Borodin, Allan, Ran El-yaniv, and Vincent Gogan. "Can We Learn to Beat the Best Stock." Advances in Neural Information Processing Systems. 2003.

[31] Borodin, A., El-Yaniv, R. and Gogan, V., 2004. Can we learn to beat the best stock. Journal of Artificial Intelligence Research, pp.579-594.

[32] Das, P. and 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.

[33] Helmbold, D.P., Schapire, R.E., Singer, Y. and Warmuth, M.K., 1998. OnLine Portfolio Selection Using Multiplicative Updates. Mathematical Finance, 8(4), pp.325-347.

[34] Agarwal, A., Hazan, E., Kale, S. and Schapire, R.E., 2006, June. Algorithms for portfolio management based on the newton method. In Proceedings of the 23rd international conference on Machine learning (pp. 9-16). ACM.

[35] Hazan, E. and Seshadhri, C., 2009, June. Efficient learning algorithms for changing environments. In Proceedings of the 26th annual international conference on machine learning (pp. 393-400). ACM.

[36] Herbster, M. and Warmuth, M.K., 1998. Tracking the best expert. Machine Learning, 32(2), pp.151-178.

[37] Helmbold, D.P., Schapire, R.E., Singer, Y. and Warmuth, M.K., 1997. A comparison of new and old algorithms for a mixture estimation problem. Machine Learning, 27(1), pp.97-119.

[38] Birge, J.R. and Louveaux, F., 2011. Introduction to stochastic programming. Springer Science & Business Media.

[39] Gaivoronski, A.A. and Stella, F., 2000. Stochastic nonstationary optimization for finding universal portfolios. Annals of Operations Research, 100(1-4), pp.165-188.

[40] Gaivoronski, A.A. and Stella, F., 2003. On-line portfolio selection using stochastic programming. Journal of Economic Dynamics and Control, 27(6), pp.1013-1043.

[41] Hazan, E. and Arora, S., 2006. Efficient algorithms for online convex optimization and their applications. Princeton University.

[42] Hazan, E., Kalai, A., Kale, S. and Agarwal, A., 2006. Logarithmic regret algorithms for online convex optimization. In Learning theory (pp. 499-513). Springer Berlin Heidelberg.

[43] Hazan, E., Agarwal, A. and Kale, S., 2007. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2-3), pp.169-192.

[44] Zinkevich, M., 2003. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning. ACM.

[45] Campbell, J.Y., Lo, A.W.C. and MacKinlay, A.C., 1997. The econometrics of financial markets (Vol. 2, pp. 149-180). Princeton, NJ: princeton University press.

[46] Cont, R., 2001. Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance 1(2), pp.223–236.

[47] Bondt, W.F. and Thaler, R., 1985. Does the stock market overreact?. The Journal of finance, 40(3), pp.793-805.

[48] Lo, A.W. and MacKinlay, A.C., 1990. When are contrarian profits due to stock market overreaction?. Review of Financial studies, 3(2), pp.175-205.

[49] Poterba, J.M. and Summers, L.H., 1988. Mean reversion in stock prices: Evidence and implications. Journal of financial economics, 22(1), pp.27-59.

[50] Officer, R.R., 1972. The distribution of stock returns. Journal of the american statistical association, 67(340), pp.807-812.

[51] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. and Singer, Y., 2006. Online passive-aggressive algorithms. The Journal of Machine Learning Research, 7, pp.551-585.

[52] Shalev-Shwartz, S., Crammer, K., Dekel, O. and Singer, Y., 2003. Online passive-aggressive algorithms. In Advances in neural information processing systems (p. None).

[53] Vardi, Y. and Zhang, C.H., 2000. The multivariate L1-median and associated data depth. Proceedings of the National Academy of Sciences, 97(4), pp.1423-1426.

[54] Weiszfeld, E., 1937. Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Math. J, 43(355-386), p.2.

[55] Weber, A. and Pick, G., 1909. Über den Standort der Industrien: Reine Theorie des Standorts. JCB Mohr.

[56] Boyd, S. and Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

[57] Li, B., HOI, C.H., Zhao, P. and Gopalkrishnan, V., 2011. Confidence weighted mean reversion strategy for on-line portfolio selection. In JMLR: Workshop and Conference Proceedings (Vol. 15, p. 434).

[58] Crammer, K., Dredze, M. and Kulesza, A., 2009, August. Multi-class confidence weighted algorithms. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2 (pp. 496-504). Association for Computational Linguistics.

[59] Crammer, K., Dredze, M. and Pereira, F., 2009. Exact convex confidence-weighted learning. In Advances in Neural Information Processing Systems (pp. 345-352).

[60] Dredze, M., Crammer, K. and Pereira, F., 2008, July. Confidence-weighted linear classification. In Proceedings of the 25th international conference on Machine learning (pp. 264-271). ACM.

[61] Dredze, M., Kulesza, A. and Crammer, K., 2010. Multi-domain learning by confidence-weighted parameter combination. Machine Learning, 79(1-2), pp.123-149.

[62] Li, B. and Hoi, S.C., 2012. On-line portfolio selection with moving average reversion. In Proceedings of the 29th international conference on Machine learning. ACM.

[63] Duchi, J., Shalev-Shwartz, S., Singer, Y. and Chandra, T., 2008, July. Efficient projections onto the l 1-ball for learning in high dimensions. In Proceedings of the 25th international conference on Machine learning (pp. 272-279). ACM.

[64] http://www.investopedia.com

[65] Kalai, A. and Vempala, S., 2003. Efficient algorithms for universal portfolios. The Journal of Machine Learning Research, 3, pp.423-440.

[66] Aldridge, I., 2009. High-frequency trading: a practical guide to algorithmic strategies and trading systems (Vol. 459). John Wiley and Sons.

Acknowledgement

I would to express my deepest gratitude to my adviser, Prof. Aviv Segev. I was fortunate to have him as my adviser who gave me full freedom to pick research area and topic for my thesis. During last 2 years, I picked many fields and topics for my research but dropped after few weeks or months. His ever-encouraging and patient nature always helped me to overcome those situations. I hope one day, I could also learn to have patience and optimism like he has.

Among other people, I would like to thank Prof. Mun Yong Yi for his encouragement and cooperation, especially in my first year at KAIST. When my adviser Prof. Segev was at the MIT during my first year at KAIST, he was like my second adviser, although unofficial or undocumented. I had a little rough time during my 1^{st} semester due to dramatic lifestyle changes at KAIST. At the time, Prof. Yi always offered his help and I am deeply indebted to that. Besides, taking his course KSE612 taught me unique perspective about the human world. It remains one of my favourite courses to the date.

My current work involves application of machine learning. I didn't have any prior experience of working in machine learning field until I took KSE624, a course which is entirely discussion and projects oriented, offered by Prof. Uichin Lee. That was my first introduction with machine learning field. I would to like to thank him for offering that wonderful course.

There are many other people who played direct or indirect role in my life, here at KAIST and back home in India, which has led me to this moment of finishing my Master's degree. My friends at KSE especially, Hyun-Gook Kang, Hyunwoo Je, Francisco J. Salgado Gonzalez, Leksikov Sergey and Sukhwan Jung played significant role during my time at KAIST in one way or other. I shall be always indebted to them for their kindness and support.

Curriculum Vitae

Name	:	Abhishek Kumar
Date of Birth	:	December 25^{th}, 1993
E-mail	:	abhishek@kaist.ac.kr

Educations

2010 – 2014	National Institute of Technology Patna, India Bachelor of Technology, Computer Science & Engineering
2014 – 2016	Korea Advanced Institute of Science and Technology (KAIST) Master of Science, Graduate School of Knowledge Service Engineering