Sequential Asset Ranking within Nonstationary Time Series

Gabriel Borrageiro*, Nick Firoozye and Paolo Barucca
Department of Computer Science, University College London, 66-72 Gower Street, London, WC1E 6BT, London, United Kingdom.

*Corresponding author. E-mail: gabriel.borrageiro.20@ucl.ac.uk; Contributing authors: n.firoozye@ucl.ac.uk; p.barucca@ucl.ac.uk

Abstract
Financial time series are both autocorrelated and nonstationary, presenting modelling challenges that violate the independent and identically distributed random variables assumption of most regression and classification models. The prediction with expert advice framework makes no assumptions on the data-generating mechanism yet generates predictions that work well for all sequences, with performance nearly as good as the best expert with hindsight. We conduct research using S&P 250 daily sampled data, extending the academic research into cross-sectional momentum trading strategies. We introduce a novel ranking algorithm from the prediction with expert advice framework, the naive Bayes asset ranker, to select subsets of assets to hold in either long-only or long/short portfolios. Our algorithm generates the best total returns and risk-adjusted returns, net of transaction costs, outperforming the long-only holding of the S&P 250 with hindsight. Furthermore, our ranking algorithm outperforms a proxy for the regress-then-rank cross-sectional momentum trader, a sequentially fitted curds and whey multivariate regression procedure.

Keywords: financial time series, online learning, prediction with expert advice, learning to rank, cross-sectional systematic strategies
1 Introduction

There are countless practical examples in machine learning where one would like to combine the predictions of a committee of experts to solve a particular forecasting challenge. A traditional form of offline (batch) learning is ensemble learning, which arguably includes neural networks with their committees of hidden processing units and certainly includes boosting and bagging algorithms. Our particular modelling interest is with financial time series, which are typically nonstationary. Nonstationarity implies variable means and variances that adapt over time and violates the independent and identically distributed random variables assumption of most regression and classification models. As such, we require approaches that adopt sequential optimisation methods to model financial data, preferably approaches that make little or no assumptions on the data generating process for the time series. The prediction with expert advice framework is a multidisciplinary area of research suited to predicting sequences sequentially, where statistical distribution assumptions are not made and are shown to minimise the regret with respect to the best available expert with hindsight. This framework provides a suitable way to model and forecast financial time series and are well-suited portfolio selection problems.

The main result of this paper is the introduction of a novel ranking algorithm, the naive Bayes asset ranker, which we use to target long-only and long/short positions in constituents of the S&P 250 index. Our algorithm is optimised sequentially, which allows it to cope with the changing interactions and behaviours of the S&P 250 constituents. Furthermore, our algorithm outperforms a strategy that would hold the long-only S&P 250 index with hindsight, despite the index appreciating by 88.2% during the test period. Our algorithm can generate excess total and risk-adjusted returns by ranking the probability that individual constituents in the index will generate higher returns than the remaining constituents. Our algorithm also outperforms a regress-then-rank cross-sectional momentum trader baseline, a sequentially fitted curds and whey multivariate regression procedure.

Our paper is organised as follows. Section 2 begins with the rationale for online learning in financial time series prediction, describes the prediction with expert advice modelling framework, moves on to online financial portfolio selection problems, incuding the ranking of constituents within portfolios of assets, and completes with an introduction to naive Bayes ranking. Section 3 presents our naive Bayes asset ranking algorithm. In section 4, we describe the design of the experiments we conduct, including details of the regress-then-rank baseline model we use, the curds and whey multivariate regression model. The section completes with a description of the results. Section 5 provides a discussion of the results and the implications for modelling financial time series. The paper concludes with remarks section 6.
2 Preliminaries

In this section, we introduce various concepts central to the algorithm we devise in section 3. We begin by arguing the case for online learning in financial time series prediction and describe the research area of prediction with expert advice, which leads to a review of the academic literature on online financial portfolio selection and the ranking of portfolios of assets. Finally, the section completes with a review of naive Bayes ranking.

2.1 The Rationale for Online Learning

Financial time series exhibit the characteristics of high serial autocorrelation and nonstationarity. Merton (1976) models the dynamics of financial assets as a jump-diffusion process, which implies that financial time series should observe small changes over time, so-called continuous changes, as well as occasional jumps. One approach for coping with nonstationarity is learning and optimising prediction models sequentially. The impact of high serial autocorrelation on time series where forecasts are made using regression-based models is that the $R^2$ of model fit will be spuriously high (Granger and Newbold, 1974). The impact of nonstationarity can be more severe in that models fitted offline on training data generalise poorly on hitherto unseen test data. One way to identify nonstationarity in time series is through the Dickey-Fuller test (Dickey and Fuller, 1979). Denote the autocovariance function as

$$\gamma(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu)(x_t - \mu)].$$

A process is said to be strictly stationary if the probabilistic behaviour of a collection of values $x_1, x_2, ..., x_k$ is identical to that of a time shifted set $x_{1+h}, x_{2+h}, ..., x_{k+h}$. It is said to be weakly stationary if

1. Its mean is constant and does not depend on time.
2. Its variance is finite and its autocovariance function $\gamma(s, t)$ depends only on $s$ and $t$ and only through their difference $|s - t|$.

Define the autoregressive(1) model with a drift term

$$\Delta y_t = \theta_0 + \theta_1 y_{t-1} + \epsilon_t.$$ 

If $\theta_1 = 0$, the time series contains a unit root, implying nonstationarity. The Dickey-Fuller critical values allow the researcher to accept or reject the null hypothesis $H_0 : \theta_1 = 0$. The standard regression t-values for the regression parameter estimates in the above model can be compared against these critical values. In section 4, we experiment with approximately the last seven years of daily sampled S&P 250 data. When running the Dickey-Fuller test against the closing prices, 83.6% of the time series are considered nonstationary. When running the test against daily returns of the form $\text{price}_t/\text{price}_{t-1} - 1$, 100% of the time series are now considered stationary. Naively one could now assume
that it is safe to perform any modelling on the daily returns, and nonstationarity will no longer present a problem. A simple experiment calls this view into question. Table 1 shows the output of the following procedure:

1. For each constituent of the S&P 250, construct daily returns using the closing prices.
2. Separate the daily samples into groups by calendar month.
3. Run the Levene test (Levene, 1960) to check if the daily-sampled data separated into monthly groups have equal variances, also known as homogeneity of variance. Levene’s test is an alternative to the Bartlett test (Bartlett, 1937) and is less sensitive than the Bartlett test to departures from normality.
4. Conduct a two-sample t-test for equal means (Snedecor and Cochran, 1989) where the current daily mean is compared against the past month’s daily mean.
5. For both tests, set the significance level $\alpha = 0.05$.

The results for the Levene test show that the null hypothesis that the batch variances are all equal can be rejected in 100% of all cases. For the two-sample t-test for equal means, the results indicate that the null hypothesis $H_0 : \mu_t = \mu_{t-1}$ can be rejected 8.1% of the time, which is evidence of nonstationarity even with daily returns. Figure 1 runs the same test, except the statistics being compared are shifted further apart in months. The curve plotted through the averages by the shift in months is the lowess curve (Cleveland, 1979). There is evidence that the null hypothesis of equal means can be rejected with a higher probability as the distance in time between the statistics increases. The traditional approach of training a model offline is likely to result in models that generalise poorly, as the statistical properties of the data are changing over time. Thus, we see firm evidence for the need to apply some form of sequential optimisation when modelling these time series.

| count | mean | var  | test stat | dof  | cv   | reject $H_0$ |
|-------|------|------|-----------|------|------|--------------|
| count | 20750| 20750| 20750     | 20444| 20444| 20444        |
| mean  | 20.819| 0.000| 0.001     | -0.024| 34.797| 1.691        |
| std   | 1.622| 0.007| 0.003     | 0.972| 5.254| 0.009        |
| min   | 12   | -0.112| 0.000     | -4.032| 11.261| 1.682        |
| 25%   | 20   | -0.003| 0.000     | -0.687| 31.525| 1.685        |
| 50%   | 21   | 0.000| 0.000     | -0.035| 36.241| 1.688        |
| 75%   | 22   | 0.003| 0.001     | 0.628| 39.027| 1.695        |
| max   | 23   | 0.151| 0.102     | 3.853| 42.000| 1.792        |
2.2 Prediction with Expert Advice

Prediction with expert advice is a multidisciplinary research area concerned with predicting individual sequences. Whereas standard statistical approaches to forecasting impose a probabilistic assumption on the data-generating mechanism, the prediction with expert advice framework makes no such assumptions yet generates predictions that work well for all sequences, with performance nearly as good as the best expert with hindsight (Cesa-Bianchi and Lugosi, 2006). The basic structure of problems in this context is encapsulated in algorithm 1, adapted from Rakhlin and Sridharan (2014).

Algorithm 1: Sequential prediction with an adaptive environment

// iterate over each time step
1 for \( t \leftarrow 1 \) to \( n \) do
2    The learner chooses the set of predictions \( \hat{y}_t \in D \), where \( D \) is the decision space.
3    Nature simultaneously chooses an outcome \( z_t \in Z \), where \( Z \) is the adversarial outcome space.
4    The player suffers a loss \( \ell(\hat{y}_t, z_t) \) and both players observe \((\hat{y}_t, z_t)\).

Perhaps the most well-known algorithm within this framework is the weighted majority algorithm of Littlestone and Warmuth (1994). The authors
study the construction of prediction algorithms where the learner faces a sequence of trials, with predictions to be made at the end of each trial, making as few mistakes as possible. They are interested in cases where the learner believes that some experts will perform well, but the learner does not know which ones specifically. A simple and effective method, based on weighted voting, is introduced for constructing a compound algorithm in such a circumstance. They show that this algorithm is robust in the presence of errors in the data and discuss various versions of the algorithm, proving mistake bounds for them that are closely related to the mistake bounds of the best algorithms of the pool. Finally, given a sequence of trials, if there is an algorithm in the pool \( A \) that makes at most \( m \) mistakes, then the weighted majority algorithm will make at most \( c(\log|A| + m) \) mistakes on that sequence, where \( c \) is a fixed constant.

If one were to criticise the weighted majority algorithm, the criticism would be directed at applying the algorithm to nonstationary data. For example, in section 3 of their paper Littlestone and Warmuth detail a modification to their base algorithm, which ensures that the weight assigned to the individual experts never dips below \( \gamma/|A| \), which is the learning rate divided by the number of experts. This fixed, minimum threshold weight is somewhat rudimentary. Our preference is for algorithms that assign more weight to experts that are now performing well that might have performed less well previously. Herbster and Warmuth (1998) ameliorate the problem of prediction of sequences in the nonstationary setting by introducing an algorithm that determines the best experts for segments of the individual sequences. When the number of segments is \( k + 1 \) and the sequence is of length \( n \), they can bound the additional loss of their algorithm over the best partition by \( O(k \log|A| + k \log n/k) \).

An Achilles heel of the algorithm is that knowledge is required a priori of the optimal number of segments \( k \) corresponding to the periods when new best experts are required. Freund et al (1997) study online learning algorithms that combine the predictions of several experts who belong to the multiplicative weights family of algorithms. They apply their methods to the problem of predicting in a model in which the best expert may change with time. They derive a specialist-based algorithm for this problem that is as fast as the best-known algorithm of Herbster and Warmuth (1998) and achieves almost as good a loss bound. However, unlike Herbster and Warmuth, Freund et al’s algorithm does not require prior knowledge of the length of the sequence and the number of switches.

### 2.3 Online Financial Portfolio Selection

The prediction with expert advice framework is valuable in sequential or online portfolio selection. Helmbold et al (1998) present an online investment algorithm that achieves almost the same wealth as the best constant-rebalanced portfolio determined in hindsight from the actual market outcomes. The algorithm employs a multiplicative update rule derived using a framework introduced by Kivinen and Warmuth (1995). They test the performance of
their algorithm on actual stock data from the New York Stock Exchange accumulated during 22 years. On these data, their algorithm outperforms the best single stock with hindsight as well as Cover’s universal portfolio selection algorithm (Cover, 1991).

Singer (1998) notes that the earlier work into online portfolio selection algorithms which are competitive with the best constant rebalanced portfolio determined in hindsight (Cover, 1991; Helmbold et al, 1998; Cover and Ordentlich, 1996), employ the assumption that high yield returns can be achieved using a fixed asset allocation strategy. However, stock markets are nonstationary, and in many cases, the return of a constant rebalanced portfolio is much smaller than the return of an ad-hoc investment strategy that adapts to changes in the market. In his paper Singer presents an efficient portfolio selection algorithm that can track a changing market and describes a simple extension of his algorithm for including transaction costs. Finally, he provides a simple analysis of the competitiveness of the algorithm and evaluates its performance on actual stock data from the New York Stock Exchange accumulated during 22 years. On this data, his algorithm outperforms all the algorithms referenced above, with and without transaction costs.

2.4 Learning to Rank Portfolios of Assets

Related to the issue of portfolio selection is the ranking of a subset of assets to buy or trade in a long/short portfolio. Poh et al (2021) apply learning to rank algorithms which are primarily designed for natural language processing, to cross-sectional momentum trading strategies. Cross-sectional strategies mitigate some of the risk associated with wider market moves by buying the top \( \alpha \)-percentile of strategies with the highest expected future returns and selling the bottom \( \alpha \)-percentile of strategies with the lowest expected future returns. Classical approaches that rely on the ranking of annualised returns (Jegadeesh and Titman, 1993), or more recent regress-then-rank approaches (Wang and Rasheed, 2018; Kim, 2019; Gu et al, 2020). Poh et al (2021) take a different approach using pair-wise learning to rank algorithms such as RankNet (Burges et al, 2005) and LambdaRank (Burges, 2010). Overall, on experiments they conduct on monthly-sampled Center for Research in Security Prices 2019 data, the learning to rank algorithms achieve higher total and risk-adjusted returns than the traditional regress-then-rank approaches. In terms of assessing the limitations of the work, experimenting with daily sampled data would probably resonate more with the finance community, as this facilitates the possibility of a more active portfolio management style. Grinold and Kahn (2019) demonstrate how active portfolio managements generally outperforms passive portfolio management and exhibit quantitative procedures to do so. However, working with higher frequency sampled data presents further challenges such as transaction cost mitigation, slower training times for the learning to rank algorithms and perhaps more crucially, a decrease in signal and increase in noise from the financial time series themselves.
2.5 Naive Bayes Ranking

In the following section, we introduce our naive Bayes ranker for portfolios comprised of financial assets. First, however, we perform a literature review of naive Bayes ranking. The phrase "naive Bayes ranking" has broad meaning, with different implementations in various contexts. At the core of the idea is the naive Bayes classifier

\[ P(y_c|x) = \frac{P(x|y_c)P(y_c)}{\sum_j P(x|y_j)P(y_j)}, \]

which predicts that \( y \) takes on a label value of \( c \) if this posterior probability is highest. The firm (naive) independence assumptions between the features \( x \) means that the probabilities can be modelled iteratively and inexpensively. Zhang and Su (2004) study the general performance of naive Bayes in ranking. They use the area under the receiver operating characteristics curve (Provost and Fawcett, 1997), also known as the AUC, to evaluate the quality of rankings generated by a classifier. Finally, they discuss how the AUC is a good metric for evaluating the quality of classifiers averaged across all possible probability thresholds. Furthermore, they show that for binary classification, AUC is equivalent to the probability that a randomly chosen example of class \(-\) will have a more negligible estimated probability of belonging to class \(+\) than a randomly chosen example of class \(+\). Thus, the AUC is a measure of the quality of ranking. The AUC is created by plotting the true-positive rate against the false-positive rate at various threshold settings and has a maximum value of one if no positive example precedes any negative example. Overall they find that their naive Bayes ranker evaluated using the AUC outperforms C4.4 on various datasets, with C4.4 being the state-of-the-art decision-tree algorithm for ranking (Provost and Domingos, 2003).

Flach and Matsubara (2007) consider binary classification tasks, where a ranker sorts a set of instances from highest to lowest expectation that the instance is positive. They propose a lexicographic ranker, LexRank, whose rankings are derived not from scores but a simple ranking of attribute values obtained from the training data. Using the odds ratio to rank the attribute values, they obtain a restricted version of the naive Bayes ranker. They systematically develop the relationships and differences between classification, ranking, and probability estimation, which leads to a novel connection between the Brier score (Brier, 1950) and AUC curves. Combining LexRank with isotonic regression (Zadrozny and Elkan, 2001), which derives probability estimates from the AUC convex hull, results in the lexicographic probability estimator LexProb. Both LexRank and LexProb are empirically evaluated on various datasets and are highly effective.
3 The Naive Bayes Asset Ranker

This section introduces our novel ranking algorithm, the Naive Bayes Asset Ranker. The algorithm can be used to sequentially rank an array of experts via a performance measure such as forecasted returns, profit and loss (pnl) or risk-adjusted returns such as the Sharpe or information ratio. Assume for the moment that the algorithm is presented with a set of \( d \) pnels. The goal here is to select a subset of experts \( 1 \leq k \leq d \) such that the reward of the \( k \) experts is expected to be the highest; this is achieved by computing the sequential posterior probability that expert \( j \in 1, \ldots, d \) is ranked higher than each of the \( d \) experts. A crucial attribute of the algorithm is that this posterior probability is computed with exponential decay using parameter \( 0 < \tau \leq 1 \), allowing experts who performed poorly previously and now perform well to be selected with weights that are not too close to zero; such behaviour is crucial for financial time series, which are nonstationary.

In section 4.1, we provide algorithmic details of a multivariate regression procedure, the curds and whey model, which is a suitable way of predicting several response variables from the same set of explanatory variables. Furthermore, by taking advantage of the correlations between the response variables, the model can reduce mean-square prediction error compared to the more common procedure of doing individual regressions of each response variable on the shared set of predictor variables. Finally, we aggregate the one-step-ahead forecasts of the curds and whey procedure and provide them to our naive Bayes asset ranker when conducting experiments on daily returns of S&P 250 data.
Algorithm 2: Naive Bayes Asset Ranker

Input: $r_t$, $\tau$

// $r_t \in \mathbb{R}^d$, a time $t$ performance measure vector we wish to track, such as forecasted returns or realised pnls
// $0 < \tau \leq 1$, an exponential decay constant

Initialise: $R_t$, $p_t$, $q_t$

// $R_0 \in 0^{d \times d}$, a $d \times d$ zero matrix
// $p_0 \in (1/d)^d$, the posterior probability that each element within it is ranked higher than the others, with elements initialised to $1/d$
// $q_0 \in 0^d$, the likelihood that each element within it is ranked higher than the others

Output: $z_t = \text{argSort}(p_t)$

// $\text{argSort}(.)$ returns the indices that would sort an array from largest to smallest value

1. for $j \leftarrow 1$ to $d$ do
   2. $s_t = I(r_t \geq r_t)$
      // $I(.)$ is equal to 1 if the condition is true, or 0 otherwise
      // The comparison is made element-wise on the vector $s_t$
   3. $R_t[j] = \tau R_t[j] - 1 + (1 - \tau) s_t$
      // : $j$ denotes all row entries in the $j^{th}$ column
4. for $j \leftarrow 1$ to $d$ do
   5. $\bar{r}_t[j] = \frac{1}{d} \sum_{i=1}^{d} R_t[ij]$
6. for $j \leftarrow 1$ to $d$ do
   7. $q_t[j] = \frac{\bar{r}_t[j]}{\sum_{i=1}^{d} \bar{r}_i}$
   8. $p_t = \tau p_{t-1} + (1 - \tau) q_t$
   9. $z_t = \text{argSort}(p_t)$

// Denote $z_t[j] = j*$ and assume that a subset of experts $1 \leq k \leq d$ is used, the test time ensemble return is

$r_{t+1}[j\ast] / \sum_{i=1}^{k} p_t[i\ast]$

4 The Research Experiment

We begin with a description of the research experiment design, including details of two experiments using the same fixed data. In the first experiment, we train and test sequentially using a regress-then-rank multivariate regression model to hold a subset of long-only positions from the S&P250 index. Additionally, we optimise a novel online ranking model from the prediction with expert advice framework to target a subset of long-only positions. In the second experiment, the same models are permitted now to hold a basket of long/short positions.
4.1 Experiment Design

We extract daily sampled S&P 250 data from Refinitiv. Designate the mid-price for the $d$th predictor as

$$ \text{mid}_t^{[d]} = 0.5 \times (\text{bid}_t^{[d]} + \text{ask}_t^{[d]}), $$

whose associated return is

$$ \text{ret}_t^{[d]} = \text{mid}_t^{[d]} / \text{mid}_{t-1}^{[d]} - 1. $$

These returns are stored in a predictor vector

$$ x_t = \begin{bmatrix} 1, \text{ret}_t^{[1]}, ..., \text{ret}_t^{[d]} \end{bmatrix}. $$

Represent the vector of one-step-ahead target returns that we want to forecast as

$$ y_{t+1} = \begin{bmatrix} \text{ret}_{t+1}^{[1]}, ..., \text{ret}_{t+1}^{[d]} \end{bmatrix}. $$

We wish to learn sequentially the multivariate mapping

$$ p(y_{t+1}|\Theta_t, x_t) = \mathcal{N}(y_{t+1}|\Theta_t x_t, \Sigma_t), $$

which is a standard regression problem. The curds and whey procedure due to Breiman and Friedman (1997) is a suitable way of predicting several response variables from the same set of explanatory variables. The method takes advantage of the correlations between the response variables to improve predictive accuracy compared with the usual procedure of doing individual regressions of each response variable on the shared set of predictor variables. The basic version of the procedure begins with the usual multivariate ridge regression

$$ \Theta = (X^T X + \lambda I_d)^{-1} X^T Y, $$

and estimates a shrinkage matrix

$$ \hat{Y} = X\Theta $$

$$ \Phi = (Y^T Y + \lambda I_q)^{-1} Y^T \hat{Y}. $$

At test time, the vector of forecasts is

$$ \tilde{y}_t = \Phi \hat{y}_t = \Phi \Theta x_t. $$

Breiman and Friedman also derive estimates of the matrix $\Phi$ that take the form $\Phi = T^{-1} D T$ where $T$ is the $q \times q$ matrix whose rows are the response canonical co-ordinates and $D$ is a diagonal ‘shrinking’ matrix. Given
the nonstationary nature of the data we model, our particular interest is in sequential optimisation. Thus we combine the curds and whey procedure with exponentially weighted recursive least-squares (ewrls) updating. This is shown in algorithm 3.

Algorithm 3: the curds and whey procedure with ewrls updating

Require: \( \lambda \) // the ridge penalty
0 \( \ll \tau < 1 \) // an exponential forgetting factor
Initialise: \( \Theta = 0^{(d+1) \times (d+1)} \), \( P = I_{d+1} / \lambda \), \( Q = I_d / \lambda \)
// \( I_{d+1} \) denotes the identity matrix of size \( d + 1 \)
Input: \( x_t \in \mathbb{R}^{d+1} \) // the inputs/predictors vector at time \( t \)
\( y_t \in \mathbb{R}^d \) // the targets/responses vector at time \( t \)
Output: \( \tilde{y}_t \) // forecast one-step-ahead returns vector

1. \( r_t = 1 + x_t^T P_{t-1} x_{t-1} / \tau \)
2. \( k_t = P_{t-1} x_{t-1} / (r_t \tau) \)
3. \( \Theta_t = \Theta_{t-1} + k_t (y_t - \Theta_{t-1} x_{t-1})^T \)
4. \( P_t = P_{t-1} / \tau - k_t k_t^T r_t \)
5. \( P_t = P_{t-1} / \tau \) // variance stabilisation
6. \( \hat{y}_t = \Theta_t x_t \)
7. \( s_t = 1 + y_t^T Q_{t-1} y_{t-1} / (s_t \tau) \)
8. \( m_t = Q_{t-1} y_{t-1} / s_t \)
9. \( \Phi_t = \Phi_{t-1} + m_t (\hat{y}_t - \Phi_{t-1} y_{t-1})^T \)
10. \( Q_t = Q_{t-1} / \tau - m_t m_t^T s_t \)
11. \( Q_t = Q_{t-1} / \tau \)
12. \( \tilde{y}_t = \Phi_t \hat{y}_t \)

We use the forecasts of algorithm 3 as the basis for taking risk in a subset of constituents in the S&P 250 index. Specifically, the top decile of forecasts \( \tilde{y}_t \) by size are bought in a long-only position experiment, and the top decile of forecasts are bought, and the bottom decile of forecasts are sold in a long/short position experiment. The long-only position constraint is of interest to mutual funds or exchange-traded funds where this constraint is applied. The long/short portfolio is of interest to the broader hedge fund community and is an example of cross-sectional momentum trading. If there are \( k \) assets in the relevant decile, each constituent’s weight of \( 1/k \) is applied.

A second forecaster that we consider is the naive Bayes asset ranker, algorithm 2, applied to the one-step-ahead forecasts of algorithm 3. Assuming there are \( k \) assets in the top decile that we wish to go long, the naive Bayes asset ranker assigns a weight to the \( j^{th} \) constituent (1 \( \leq j \leq k \)) of

\[
p_t^{[j\ast]} / \sum_{i\ast=1}^{k} p_t^{[i\ast]}.
\]
For short positions, assuming there are $k$ assets we wish to go short, the weight assigned to the $j^{th}$ constituent $(d - k \leq j \leq d)$ is

$$
\left(1 - p_i^{[j]} \right) / \left( \sum_{i=d-k}^{d} 1 - p_i^{[i]} \right).
$$

We must also consider execution costs. We force the curds and whey model and the naive Bayes asset ranker to trade as a price taker, meaning that the models incur a cost equal to half the bid/ask spread times the change in absolute position. Furthermore, as these data are sampled daily, any portfolio rebalancing is applied at most once a day, at the close of trading circa 4 pm EST. Finally, the hyperparameters that are set for models in this experiment are exponential decay $\tau = 0.999$ for algorithms 2 and 3, and ridge penalty $\lambda = 1$ for algorithm 3. There is no particular reason for this choice of hyperparameters other than setting them by default to these values. Some performance gains could be made by estimating these hyperparameters via a cross-validation procedure, but this would reduce the test set size, so we avoid performing cross-validation here.

4.2 Results

Two experiments using the same data are considered. In the first experiment, we consider long-only positions, and in the second, we consider cross-sectional momentum long/short positions.

4.2.1 Long Only Positions

With hindsight, holding an equal weighting of constituents in the S&P 250 would earn a total return of 88.2%, implying a compound annual growth rate of 14.3%. The annualised Sharpe ratio is 0.74, which implies that the probability that the annualised return will be greater than zero is 69.7%. The maximum drawdown is 52.9% which occurred in the March 2020 Covid-19 induced sell-off. The total return to maximum drawdown coverage is $1.67 \times$. The final statistic to consider is the win ratio, which is positive pnl days to all trading days; this comes in at 53.7%. The long-only curds and whey procedure achieves a test-set total return of 135.4%, suggesting a compound annual growth rate of 19.8%. The annualised Sharpe ratio is 0.92, intimating a probability that the annualised return will be greater than zero, of 73.8%. Finally, the total return to maximum drawdown coverage is $2.4 \times$; thus, this model has outperformed holding the long-only S&P 250 with hindsight. The naive Bayes asset ranker has the highest total return and risk-adjusted return. It achieves a total return of 156.7%, intimating a compound annual growth rate of 22.1%. The annualised Sharpe ratio is 1.17, implying a probability of annualised return greater than zero, of 79.8%. Finally, the total return to maximum drawdown coverage is $3.2 \times$. 
Sequential Asset Ranking within Nonstationary Time Series

Fig. 2 total return for the curds and whey regression model and naive Bayes asset ranker, where long-only positions are permitted

Table 2 return statistics for the curds and whey multivariate regression and naive Bayes asset ranker, where only long positions are permitted

|                | long only S&P 250 | long only Curds and Whey | long only Naive Bayes Asset Ranker |
|----------------|-------------------|--------------------------|-----------------------------------|
| days           | 1726              | 1726                     | 1726                              |
| mean           | 0.001             | 0.001                    | 0.001                             |
| std            | 0.013             | 0.016                    | 0.015                             |
| min            | -0.126            | -0.125                   | -0.142                            |
| 25%            | -0.005            | -0.008                   | -0.006                            |
| 50%            | 0.001             | 0.001                    | 0.001                             |
| 75%            | 0.007             | 0.009                    | 0.008                             |
| max            | 0.097             | 0.085                    | 0.109                             |
| sum            | 0.882             | 1.354                    | 1.567                             |
| cagr           | 0.143             | 0.198                    | 0.221                             |
| sr             | 0.742             | 0.916                    | 1.167                             |
| pr(pnl>0)      | 0.697             | 0.738                    | 0.798                             |
| max dd         | 0.529             | 0.570                    | 0.495                             |
| pnl / max dd   | 1.668             | 2.375                    | 3.163                             |
| win ratio %    | 0.537             | 0.518                    | 0.532                             |
| loss ratio %   | 0.463             | 0.482                    | 0.468                             |

4.2.2 Long/Short Positions

In the experiment where we trade cross-sectional momentum with long/short positions, the curds and whey procedure achieves a test-set total return of 97.9%, implying a compound annual growth rate of 15.5%. The risk-adjusted
Performance, measured through the annualized Sharpe ratio, comes in at 0.76. This Sharpe ratio suggests that the probability that the annualized return will be greater than zero is 70.1%. The total return to maximum drawdown coverage has improved, showing $2.8\times$, reflecting a lower maximum drawdown of 34.8%. As before, the naive Bayes asset ranker has the highest total return and risk-adjusted return. It achieves a total return of 149.3%, implying a compound annual growth rate of 21.3%. Furthermore, the annualized Sharpe ratio has increased to 1.47, which intimates that the probability that the annualized return will be greater than zero is 86.5%. The total return to maximum drawdown coverage has increased to $8.6\times$, and the win ratio is also highest, at 54.1%. Figure 4 shows the long/short cumulative selection probability by sector for the naive Bayes asset ranker. We see that it draws long positions predominately from constituents in the manufacturing, energy & transportation and trade & services sectors. Additionally, it draws short positions mostly from the life sciences, trade & services and energy & transportation sectors.

Fig. 3 total return for the curds and whey multivariate regression and naive Bayes asset ranker, where long and short positions are permitted

5 Discussion

It is clear from the results that the market has appreciated during the experiments test period. Our proxy for regress-then-rank modelling, the curds and whey procedure, performs better than the long-only S&P 250 (with hindsight), irrespective of the long-only or long/short constraints. However, there is a
Table 3 Return statistics for the curds and whey multivariate regression and naive Bayes asset ranker, where long and short positions are permitted

|                | long only S&P 250 | long/short Curds and Whey | long/short Naive Bayes Asset Ranker |
|----------------|-------------------|---------------------------|-------------------------------------|
| days           | 1726              | 1726                      | 1726                                |
| mean           | 0.001             | 0.001                     | 0.001                               |
| std            | 0.013             | 0.014                     | 0.011                               |
| min            | -0.126            | -0.070                    | -0.048                              |
| 25%            | -0.005            | -0.008                    | -0.006                              |
| 50%            | 0.001             | 0.001                     | 0.001                               |
| 75%            | 0.007             | 0.009                     | 0.007                               |
| max            | 0.097             | 0.075                     | 0.068                               |
| sum            | 0.882             | 0.979                     | 1.493                               |
| cagr           | 0.143             | 0.155                     | 0.213                               |
| pr(pnl>0)      | 0.697             | 0.701                     | 0.865                               |
| max dd         | 0.529             | 0.348                     | 0.173                               |
| pnl / max dd   | 1.668             | 2.812                     | 8.636                               |
| win ratio %    | 0.537             | 0.514                     | 0.541                               |
| loss ratio %   | 0.463             | 0.486                     | 0.459                               |

deterioration in total return and risk-adjusted return when forcing the model to take short positions. The naive Bayes asset ranker performs well under both scenarios, with risk-adjusted returns highest with the long/short portfolio. How might we rationalise or explain the deterioration in the performance of the curds and whey regress-then-rank procedure and the excellent performance of the naive Bayes asset ranker? In the former case, several academic papers discuss some shortcomings with regression models compared to classification models in the financial time series prediction setting. For example, Satchell and Timmermann (1995) show that regression models that typically minimise mean-square prediction error obtain worse performance than a random-walk model when forecasting daily foreign exchange rates. Furthermore, they show that the probability of correctly predicting the sign of the change in daily exchange rates can be higher for the non-linear forecasts than for the random-walk forecasts, even though the mean-square prediction error of the non-linear forecasts exceeds that of the random-walk forecasts. They conclude that the mean-square prediction error is not always an appropriate performance measure for evaluating predictions of non-linear processes.

Moody et al (1998) discuss how supervised learning techniques that are used to minimise forecast error are not guaranteed to optimise trading system performance globally. Furthermore, they state that trading system profits depend upon sequences of interdependent decisions and are thus path-dependent. Examples of path-dependent trading decisions include evaluating transactions costs, market impact and taxes, which all require knowledge of the current system state. Including information related to past decisions in the inputs to a trading system results in a recurrent decision system, requiring different optimisation from supervised optimisation techniques used for direct
Sequential Asset Ranking within Nonstationary Time Series

Fig. 4 naive Bayes asset ranker long/short selection probability by sector

forecasts. More recently, Amjad and Shah (2017) find that classical time series regression algorithms, such as auto-regressive integrated moving average models, have poor performance when forecasting Bitcoin returns. They argue that the probability distribution of the sign of future price changes can be adequately approximated from finite data, specifically classification algorithms that estimate this conditional probability distribution. We conclude by saying that if recent past data relates to near-future data that is hitherto unseen, our naive Bayes asset ranker will outperform regress-then-rank models and, at the very least, achieve a loss close to the best expert with hindsight. It will achieve this as it directly models the conditional posterior probability that a portfolio constituent will be ranked higher than the other constituents in the portfolio.

6 Conclusion

Financial time series are both autocorrelated and nonstationary, presenting modelling challenges that violate the independent and identically distributed random variables assumption of most regression and classification models. The prediction with expert advice framework makes no assumptions on the data-generating mechanism yet generates predictions that work well for all sequences, with performance nearly as good as the best expert with hindsight. We conduct research using S&P 250 daily sampled data, extending the academic research into cross-sectional momentum trading strategies. We
introduce a novel ranking algorithm from the prediction with expert advice framework, the naive Bayes asset ranker, to select subsets of assets to hold in either long-only or long/short portfolios. Our algorithm generates the best total returns and risk-adjusted returns, net of transaction costs, outperforming the long-only holding of the S&P 250 with hindsight. Furthermore, our ranking algorithm outperforms a proxy for the regress-then-rank cross-sectional momentum trader, a sequentially fitted curds and whey multivariate regression procedure.

References

Amjad M, Shah D (2017) Trading bitcoin and online time series prediction. In: Anava O, Khaleghi A, Cuturi M, et al (eds) Proceedings of the Time Series Workshop at NIPS 2016, Proceedings of Machine Learning Research, vol 55. PMLR, Barcelona, Spain, pp 1–15

Bartlett MS (1937) Properties of sufficiency and statistical tests. Proceedings of the Royal Society of London Series A, Mathematical and Physical Sciences 160(901):268–282

Breiman L, Friedman JH (1997) Predicting multivariate responses in multiple linear regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59(1):3–54

Brier GW (1950) Verification of forecasts expressed in terms of probability. Monthly Weather Review 78(1):1 – 3

Burges C, Shaked T, Renshaw E, et al (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning. Association for Computing Machinery, New York, NY, USA, ICML ’05, p 89–96

Burges CJC (2010) From RankNet to LambdaRank to LambdaMART: An overview. Tech. rep., Microsoft Research

Cesa-Bianchi N, Lugosi G (2006) Prediction, Learning, and Games. Cambridge University Press, USA

Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association 74(368):829–836

Cover T, Ordentlich E (1996) Universal portfolios with side information. IEEE Transactions on Information Theory 42(2):348–363

Cover TM (1991) Universal portfolios. Mathematical Finance 1(1):1–29
Dickey DA, Fuller WA (1979) Distribution of the estimators for autoregressive time series with a unit root. Journal of the American Statistical Association 74(366):427

Flach P, Matsubara ET (2007) A Simple Lexicographic Ranker and Probability Estimator. Springer Berlin Heidelberg

Freund Y, Schapire RE, Singer Y, et al (1997) Using and combining predictors that specialize. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing. Association for Computing Machinery, New York, NY, USA, STOC ’97, p 334–343

Granger C, Newbold P (1974) Spurious regressions in econometrics. Journal of econometrics 2:111–120

Grinold R, Kahn R (2019) Advances in Active Portfolio Management: New Developments in Quantitative Investing. McGraw-Hill Education

Gu S, Kelly B, Xiu D (2020) Empirical Asset Pricing via Machine Learning. The Review of Financial Studies 33(5):2223–2273

Helmbold DP, Schapire RE, Singer Y, et al (1998) On-line portfolio selection using multiplicative updates. Mathematical Finance 8(4):325–347

Herbster M, Warmuth MK (1998) Tracking the best expert. Mach Learn 32(2):151–178

Jegadeesh N, Titman S (1993) Returns to buying winners and selling losers: Implications for stock market efficiency. The Journal of Finance 48:65–91

Kim S (2019) Enhancing the momentum strategy through deep regression. Quantitative Finance 19(7):1121–1133

Kivinen J, Warmuth MK (1995) Additive versus exponentiated gradient updates for linear prediction. In: Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing. Association for Computing Machinery, New York, NY, USA, STOC ’95, p 209–218

Levene H (1960) Contributions to probability and statistics. Essays in honor of Harold Hotelling pp 278–292

Littlestone N, Warmuth MK (1994) The weighted majority algorithm. Information and Computation 108:212–261

Merton RC (1976) Option pricing when underlying stock returns are discontinuous. Journal of Financial Economics 3(1):125 – 144
Sequential Asset Ranking within Nonstationary Time Series

Moody J, Wu L, Liao Y, et al (1998) Performance functions and reinforcement learning for trading systems and portfolios. Journal of Forecasting 17(5-6):441–470

Poh D, Lim B, Zohren S, et al (2021) Building cross-sectional systematic strategies by learning to rank. The Journal of Financial Data Science

Provost F, Domingos P (2003) Tree induction for probability-based ranking. Machine Learning 52:199–215

Provost F, Fawcett T (1997) Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press, KDD’97, p 43–48

Rakhlin S, Sridharan K (2014) Statistical learning theory and sequential prediction. Tech. rep., MIT, sTAT928

Satchell S, Timmermann A (1995) An assessment of the economic value of nonlinear foreign exchange rate forecasts. Journal of Forecasting 14(6):477–497

Singer Y (1998) Switching portfolios. In: International Journal of Neural Systems. Morgan Kaufmann, pp 488–495

Snedecor GW, Cochran WG (1989) Statistical methods, eight edition. Iowa state University press, Ames, Iowa 1191

Wang L, Rasheed K (2018) Stock ranking with market microstructure, technical indicator and news. Proceedings on the International Conference on Artificial Intelligence (ICAI) pp 322–328

Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML ’01, p 609–616

Zhang H, Su J (2004) Naive bayesian classifiers for ranking. In: Boulicaut JF, Esposito F, Giannotti F, et al (eds) Machine Learning: ECML 2004. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 501–512