Risk-Sensitive Online Learning

Eyal Even-Dar, Michael Kearns, and Jennifer Wortman

Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104
{evendar,wortmanj}@seas.upenn.edu, mkearns@cis.upenn.edu

Abstract. We consider the problem of online learning in settings in which we want to compete not simply with the rewards of the best expert or stock, but with the best trade-off between rewards and *risk*. Motivated by finance applications, we consider two common measures balancing returns and risk: the *Sharpe ratio* [7] and the *mean-variance* criterion of Markowitz [6]. We first provide negative results establishing the impossibility of no-regret algorithms under these measures, thus providing a stark contrast with the returns-only setting. We then show that the recent algorithm of Cesa-Bianchi et al. [3] achieves nontrivial performance under a modified bicriteria risk-return measure, and also give a no-regret algorithm for a "localized" version of the mean-variance criterion. To our knowledge this paper initiates the investigation of explicit risk considerations in the standard models of worst-case online learning.

1 Introduction

Despite the large literature on online learning, and the rich collection of algorithms with guaranteed worst-case regret bounds, virtually no attention has been given to the risk incurred by such algorithms[1]. Especially in finance-related applications [4], where consideration of various measures of the volatility of a portfolio are often given equal footing with the returns themselves, this omission is particularly glaring.

The finance literature on balancing risk and return, and the proposed metrics for doing so, are far too large to survey here (see [1], chapter 4 for a nice overview). But among the two most common methods are the *Sharpe ratio* [7], and the *mean-variance (MV)* criterion of which Markowitz was the first proponent [6]. Let $r_t \in [-1, \infty]$ be the return of any given financial instrument (a stock, bond, portfolio, trading strategy, etc.) during time period t. Thus, if v_t represents the dollar value of the instrument immediately after period t, we have $v_t = (1 + r_t)v_{t-1}$. Negative values of r_t (down to -1, representing the limiting case of the instrument losing all of its value) are losses, and positive values are gains. For a sequence of returns $\boldsymbol{r} = (r_1, \ldots, r_T)$ we use $\mu(\boldsymbol{r})$ to denote the (arithmetic) mean or average value, and $\sigma(\boldsymbol{r})$ to denote the standard deviation. Then the Sharpe ratio of the instrument on the sequence is simply $\mu(\boldsymbol{r})/\sigma(\boldsymbol{r})$,

[1] A partial exception is the recent work of [3], which we analyze in our framework.

while the MV is $\mu(\boldsymbol{r}) - \sigma(\boldsymbol{r})$. (Note that the term mean-variance is slightly misleading since the risk is actually measured by the standard deviation, but we use this term to adhere to convention.) A common alternative is to use the mean and standard deviation not of the r_t but of the $\log(1 + r_t)$, which corresponds to geometric rather than arithmetic averaging of returns (see Section 2); we shall refer to the resulting measures the *geometric* Sharpe ratio and MV.

Both the Sharpe ratio and the MV are natural, if somewhat different, methods for specifying a trade-off between the risk and returns of a financial instrument. Note that if we have an algorithm (like EG) that maintains a dynamically weighted and rebalanced portfolio over K constituent stocks, this algorithm itself has a sequence of returns and thus its own Sharpe ratio and MV. A natural hope for online learning would be to replicate the kind of no-regret results to which we have become accustomed, but for regret in these risk-return measures. Thus (for example) we would like an algorithm whose Sharpe ratio or MV at sufficiently long time scales is arbitrarily close to the *best* Sharpe ratio or MV of any of the K stocks. The prospects for these and similar results are the topic of this paper.

Our first results are negative, and show that the specific hope articulated in the last paragraph is unattainable. More precisely, we show that for either the (arithmetic or geometric) Sharpe ratio or MV, any online learning algorithm must suffer *constant* regret, even when $K = 2$. This is in sharp contrast to the literature on returns alone, where it is known that zero regret can be approached rapidly with increasing T. Furthermore, and perhaps surprisingly, for the case of the Sharpe ratio the proof shows that constant regret is inevitable even for an *offline* algorithm (which knows in advance the specific sequence of returns for the two stocks, but still must compete with the best Sharpe ratio on all time scales).

The fundamental insight in these impossibility results is that the risk term in the different risk-return metrics introduces a "switching cost" not present in the standard return-only settings. Intuitively, in the return-only setting, no matter what decisions an algorithm has made up to time t, it can choose (for instance) to move all of its capital to one stock at time t, and *immediately* begin enjoying the *same* returns as that stock from that time forward. However, under the risk-return metrics, if the returns of the algorithm up to time t have been quite different (either higher or lower) than those of the stock, the algorithm pays a "volatility penalty" not suffered by the stock itself.

These strong impossibility results force us to revise our expectations for online learning for risk-return settings. In the second part of the paper, we examine two different approaches to algorithms for MV-like metrics. In the first approach, we analyze the recent algorithm of [3] and show that it exhibits a trade-off compared to the best stock under an additive measure balancing returns with variance (as opposed to standard deviation). The notion of approximation is weaker than competitive ratio or no-regret, but remains nontrivial, especially in light of the strong negative results mentioned above. In the second approach, we give a general transformation of the instantaneous rewards given to algorithms (such as EG) meeting standard returns-only no-regret criteria. This transformation permits us to incorporate a recent moving window of variance into the instantaneous rewards, yielding an algorithm competitive with a "localized" version of MV in which we are penalized only for volatility on short (compared to $\sqrt{T}$) time scales. This measure may be of independent interest.

2 Preliminaries

We denote the set of experts as integers $\mathbf{K} = \{1, \ldots, K\}$ where $K = |\mathbf{K}|$. For each expert $k \in \mathbf{K}$, we denote its *reward* at time $t \in \{1, \ldots, T\}$ as x_t^k. At each time step t, an algorithm A assigns a weight $w_t^k \geq 0$ to each expert k such that $\sum_{k=1}^{K} w_t^k = 1$. Based on these weights, the algorithm then receives a reward $x_t^A = \sum_{k=1}^{K} w_t^k x_t^k$.

There are multiple ways to define the aforementioned rewards. In a financial setting it is common to define them to be the *simple returns* of some underlying investment. Thus if v_t represents the dollar value of an investment following period t, and $v_t = (1 + r_t)v_{t-1}$ where $r_t \in [-1, \infty]$, one choice is to let $x_t = r_t$. Here negative values of r_t represent losses, while positive values represent gains.

One disadvantage of this definition is that since we are simply averaging the returns, a return of -1 — which corresponds to losing our entire investment — can be "offset" by a return of 1 — which corresponds to doubling our investment. Clearly it is odd to view these as balancing events. For this and a variety of other reasons one often wishes to consider a definition of rewards derived from geometric rather than arithmetic averaging of simple returns. The geometric average of returns $\bar{r}_{geo}$ is defined as the solution to the equation $(1 + \bar{r}_{geo})^T = \prod_{t=1}^{T}(1+r_t)$. Thus, $\bar{r}_{geo}$ represents the *fixed* rate of return yielding the equivalent T-step growth or loss of the individually varying r_t. If each time step is a year, this is often also called the *annualized* rate of return.

By taking logarithms of both sides of the above equation, it is easy to see that maximizing the geometric average of returns is equivalent to maximizing the (standard) average of the values $\log(1 + r_t)$. This suggests a second natural definition of the reward x_t as $\log(1 + r_t)$, which we call the *geometric returns*. Clearly the geometric returns are not vulnerable to the disadvantage cited above, since $r_t = -1$ gives $\log(1 + r_t) = -\infty$.

All the results presented in this paper hold for both the interpretation of rewards x_t as simple returns r_t, and for the interpretation of rewards as geometric returns $\log(1 + r_t)$. From this point on, we refer only to "rewards" and leave the choice of interpretation to the reader. We assume that daily rewards lie in the range $[-M, M]$ for some constant M. Some of our bounds may depend on M.

There is no single correct measure of volatility of rewards either. Two well-known measures that we will refer to often are variance and standard deviation. Formally, if $\bar{R}^t(k, \boldsymbol{x})$ is the average reward of expert k on the reward sequence $\boldsymbol{x}$ at time t, then

$$Var^t(k, \boldsymbol{x}) = \frac{\sum_{t'=1}^{t}(x_{t'}^k - \bar{R}^t(k, \boldsymbol{x}))^2}{t}, \qquad \sigma^t(k, \boldsymbol{x}) = \sqrt{Var^t(k, \boldsymbol{x})}$$

We define $R^t(k, \boldsymbol{x})$ to be the total reward of expert k at time t. We often abuse notation and write $R^t(k)$, $\bar{R}^t(k)$, and $\sigma^T(k)$ when $\boldsymbol{x}$ is clear from context.

Traditionally in online learning the objective of an algorithm A has been to achieve an average reward at least as good as the best expert over time, yielding results of the form

$$\max_{k \in \mathbf{K}} \bar{R}^T(k, \boldsymbol{x}) = \max_{k \in \mathbf{K}} \sum_{t=1}^{T} \frac{x_t^k}{T} \leq \sum_{t=1}^{T} \frac{x_t^A}{T} + \sqrt{\frac{\log K}{T}} = \bar{R}^T(A, \boldsymbol{x}) + \sqrt{\frac{\log K}{T}}$$

An algorithm that achieves this desired goal is often referred as a "no regret" algorithm.

Now we are ready to define two standard risk-reward balancing criteria, the Sharpe ratio [7] and the MV of expert k at time t.

$$\mathit{Sharpe}^t(k, \boldsymbol{x}) = \frac{\bar{R}^t(k, \boldsymbol{x})}{\sigma^t(k, \boldsymbol{x})}, \qquad MV^t(k, \boldsymbol{x}) = \bar{R}^t(k, \boldsymbol{x}) - \sigma^t(k, \boldsymbol{x})$$

In the following definitions we use the MV but all definitions are identical for the Sharpe ratio. We say that an algorithm has *no regret* with respect to the MV if

$$\max_{k \in \mathbf{K}} MV^T(k, \boldsymbol{x}) - \mathit{Regret}(T) \leq MV^T(A, \boldsymbol{x})$$

where $\mathit{Regret}(T)$ is a function that goes to 0 as T approaches infinity. Similarly, we can define several negative concepts. We say that an algorithm A has *constant regret* C for some constant C (that does not depend on time but may depend on M) if for any large T there exists a sequence $\boldsymbol{x}$ of expert rewards for which the following is satisfied:

$$\max_{k \in \mathbf{K}} MV^T(k, \boldsymbol{x}) > MV^T(A, \boldsymbol{x}) + C.$$

Finally, the *competitive ratio* of an algorithm A is defined as

$$\inf_{\boldsymbol{x}} \inf_{t} \frac{MV^t(A, \boldsymbol{x})}{\max_{k \in \mathbf{K}} MV^t(k, \boldsymbol{x})}$$

where $\boldsymbol{x}$ can be any reward sequence generated for K experts.

Note that for negative results it is sufficient to consider a single sequence of expert rewards for which *no* algorithm can perform well.

3 A Lower Bound for the Sharpe Ratio

In this section we show that even an offline policy cannot compete with the best expert with respect to the Sharpe ratio, even when there are only two experts. Our precise lower bound is stated in Theorem 1. The remainder of the section contains a proof of this bound.

Theorem 1. *For any $T \geq 30$, there exists an expert reward sequence $\boldsymbol{x}$ of length T such that the optimal offline algorithm has constant regret. Furthermore, on this sequence there are two points such that no algorithm can attain more than a $1 - c$ competitive ratio at both of them, for some positive constant c.*

This lower bound can be proved in a setting where there are only two experts. We start by characterizing the optimal offline algorithm and later construct a sequence on which the optimal algorithm cannot compete. This, of course, implies that no algorithm can compete. Although in general sequences can vary in each time step, the sequences used here will be more limited and will change only m times.

An **m-segment sequence** is a sequence described by expert rewards at m times, $n_1 < n_2 < ... < n_m$, such that for all $i \in \{1, \ldots, m\}$, every expert reward in the time segment $[n_{i-1}+1, n_i]$ is constant, i.e. $\forall t \in [n_{i-1}+1, n_i]$, $x_t^k = x_{n_i}^k$ for every $k \in \mathbf{K}$ where $n_0 = 0$. We say that an algorithm has a *fixed policy* in the ith segment if the weights that the algorithm places on each expert remain constant between times $n_{i-1}+1$ and n_i.

Before giving the proof of Theorem 1, we provide the following lemma, which states that the algorithm that achieves the maximal Sharpe ratio at time n_i must use a fixed policy at every segment prior to i.

Lemma 1. *Let $\boldsymbol{x}$ be an m-segment reward sequence. Let A_i^r (for $i \leq m$) be the set of algorithms that have average reward r on $\boldsymbol{x}$ at time n_i. Then the algorithm $A \in A_i^r$ with minimal standard deviation has a fixed policy in every segment prior to i. The optimal Sharpe ratio at time n_i is thus attained by an algorithm that has a fixed policy in every segment prior to i.*

The intuition behind this lemma is that switching weights within a segment can only result in higher variance without enabling an algorithm to achieve an average reward any higher than it would have been able to achieve by using a fixed set of weights in this segment. Details of the proof have been omitted due to space limitations.

With this lemma, we are ready to prove Theorem 1. We will consider one specific 3-segment sequence and show that there is no algorithm that can have competitive ratio bigger than 0.71 at both times n_2 and n_3 on this sequence. The intuition behind this construction is that in order for the algorithm to have a good competitive ratio at time n_2 it cannot put too much weight on expert 1 and has to put a significant weight on expert 2. However, putting significant weight on expert 2 prevents the algorithm from being competitive in time n_3 where it must have switched completely to expert 1 to maintain a good Sharpe ratio.

The lower bound Sharpe sequence is a 3-segment sequence composed of two experts. The three segments are of equal length. The rewards for expert 1 are .05, .01, and .05 in intervals 1, 2, and 3 respectively. The rewards for expert 2 are .011, .009, and .05. The Sharpe ratio of the algorithm will be compared to the Sharpe ratio of the best expert at times n_2 and n_3. Note that since the Sharpe ratio is a unitless measure, we could scale the rewards in this sequence by any positive constant factor and the proof would still hold.

Analyzing the sequence we observe that the best expert at time n_2 is expert 2 with Sharpe ratio 10. The best expert at n_3 is expert 1 with Sharpe ratio approximately 1.95. The remainder of the proof shows that if the average reward of the algorithm at time n_2 is "too high," then the competitive ratio at time n_2 is bad, while if the average reward at time n_2 is "too low," then the competitive ratio is bad at time n_3.

Suppose first that the average reward of the algorithm on the lower bound Sharpe sequence $\boldsymbol{x}$ at time n_2 is at least .012. The reward in the second segment can be at most .01, so if the average reward at time n_2 is $.012 + z$ where z is positive constant smaller than .018, then the standard deviation of the algorithm at n_2 is at least $.002+z$. This implies that the algorithm's Sharpe ratio is at most $\frac{.012+z}{.002+z}$, which is at most 6. Comparing this to the Sharpe ratio of 10 obtained by expert 2, we see that the algorithm can have a competitive ratio no higher than 0.6, or equivalently the algorithm's regret is at least 4.

Suppose instead that the average reward of the algorithm on $\boldsymbol{x}$ at time n_2 is less than .012. Note that the Sharpe ratio of expert 1 at time n_3 is approximately $\frac{.03667}{.018} > 1.94$. In order to obtain a bound that holds for any algorithm with average reward at most .012 at time n_2, we consider the algorithm A which has reward of .012 in every time step and clearly outperforms any other algorithm.[2] The average reward of A for the third segment must be .05 as it is the reward of both experts. Now we can compute its average and standard deviation $\bar{R}^{n_3}(A, \boldsymbol{x}) \approx 2.4667$ and $\sigma^{n_3}(A, \boldsymbol{x}) \approx 1.79$. The Sharpe ratio of A is then approximately 1.38, and we find that A has a competitive ratio at time n_3 that is at most 0.71 or equivalently its regret is at least 0.55.

The lower bound sequence that we used here can be further improved to obtain a competitive ratio of .5. The improved sequence is of the form $n, 1, n$ for the first expert's rewards, and $1+1/n, 1-1/n, n$ for the second expert's rewards. As n approaches infinity, the competitive ratio of the Sharpe ratio tested on two checkpoints at n_2 and n_3 approaches .5.

4 A Lower Bound for MV

In this section we provide a lower bound for our additive risk-reward measure, the MV.

Theorem 2. *Let A be any online algorithm. There exists a sequence $\boldsymbol{x}$ for which the regret of A with respect to the metric MV is constant.*

Again our proof will be based on specific sequences that will serve as a counterexample to show that in general it is not possible to compete with the best expert in terms of the MV. We begin by describing how these sequences are generated. Again we consider a scenario in which there are only two experts.

[2] Of course such an algorithm cannot exist for this sequence

For the first n time steps, the first expert receives at each time step a reward of 2 with probability 1/2 or a reward of 0 with probability 1/2, while at times $n+1,...,2n$ the reward is always 1. The second expert's reward is always 1/4 throughout the entire sequence. The algorithm's performance will be tested only at times n and $2n$, and the algorithm is assumed to know the process by which these expert rewards are generated.

Note that this lower bound construction is not a single sequence but is a set of sequences generated according to the distribution over the first expert's rewards. Throughout this section, we will refer to the set of all sequences that can be generated by this distribution as S. We will show by the probabilistic method that there is no algorithm that can perform well on all sequences in S at both checkpoints. In contrast to "standard" experts, there are now two randomness sources: the internal randomness of the algorithm and the randomness of the rewards.

Before delving more deeply into the details of the proof, we give a high level overview. First we will consider a "balanced sequence" in S in which expert 1 receives an equal number of rewards that are 2 and rewards that are 0. Assuming such a sequence, it will be the case that the best expert at time n is expert 2 with reward 1/4 and standard deviation 0, while the best expert at time $2n$ is expert 1 with reward 1 and standard deviation $1/\sqrt{2}$. Note that any algorithm that has average reward 1/4 at time n in this scenario will be unable to overcome this start and will have a constant regret at time $2n$. Yet it might be the case on such sequences that a sophisticated adaptive algorithm could have an average reward higher than 1/4 at time n and still suffer no regret at time n. Hence, for the balanced sequence we add the requirement that the *algorithm* is "balanced" as well, i.e. the weight it puts on expert 1 on days with reward 2 is equal to the weight it puts on expert 1 on days with reward 0.

In our analysis we show that most sequences in S are close to the balanced sequence. In particular, if the average reward of an algorithm over all sequences is less than $1/4+\delta$, for some constant δ, then by the probabilistic method there exists a sequence for which the algorithm will have constant regret at time $2n$. If not, then it can be shown that there exists a sequence for which at time n the algorithm's standard deviation will be larger than δ by some constant factor, and thus the algorithm will have regret at time n. This argument will also be probabilistic, preventing the algorithm from constantly being "lucky."

In this analysis we use a form of Azuma's inequality, which we present here for sake of completeness. Note that we cannot use standard Chernoff bound since we would like to provide bounds on the behavior of adaptive algorithms.

Lemma 2 (Azuma). *Let $\zeta_0, \zeta_1, ..., \zeta_n$ be a martingale sequence such that for each i, $1 \le i \le n$, we have $|\zeta_i - \zeta_{i-1}| \le c_i$ where the constant c_i may depend on i. Then for $n \ge 1$ and any $\epsilon > 0$*

$$Pr\left[|\zeta_n - \zeta_0| > \epsilon\right] \le 2e^{-\frac{\epsilon^2}{2\sum_{i=1}^{n} c_i^2}}$$

Now we define two martingale sequences, $y_t(\boldsymbol{x})$ and $z_t(A, \boldsymbol{x})$. The first counts the difference between the number of times expert 1 receives a reward of 2 and the number of times expert 1 receives a reward of 0 on a given sequence $\boldsymbol{x} \in S$. The second counts the difference between the weights that algorithm A places on expert 1 when expert 1 receives a reward of 2 and the weights placed on expert 1 when expert 1 receives a reward of 0. We define $y_0(\boldsymbol{x}) = z_0(A, \boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ and A.

$$y_{t+1}(\boldsymbol{x}) = \begin{cases} y_t(\boldsymbol{x}) + 1, & x^1_{t+1} = 2 \\ y_t(\boldsymbol{x}) - 1, & x^1_{t+1} = 0 \end{cases}, \quad z_{t+1}(A, \boldsymbol{x}) = \begin{cases} z_t(A, \boldsymbol{x}) + w^1_{t+1}, & x^1_{t+1} = 2 \\ z_t(A, \boldsymbol{x}) - w^1_{t+1}, & x^1_{t+1} = 0 \end{cases}$$

In order to simplify notation throughout the rest of this section, we will often drop the parameters and write y_t and z_t when A and $\boldsymbol{x}$ are clear from context.

Recall that $\bar{R}^n(A, \boldsymbol{x})$ is the average reward of an algorithm A on sequence $\boldsymbol{x}$ at time n. We denote the *expected* average reward at time n as $\bar{R}^n(A, D) = E_{\boldsymbol{x} \sim D}\left[\bar{R}^n(A, \boldsymbol{x})\right]$, where D is the distribution over rewards.

Next we define a set of sequences that are close to the balanced sequence on which the algorithm A will have a high reward, and subsequently show that for algorithms with high expected average reward this set is not empty.

Definition 1. *Let A be any algorithm and δ any positive constant. Then the set S^δ_A is the set of sequences $\boldsymbol{x} \in S$ that satisfy (1) $|y_n(\boldsymbol{x})| \le \sqrt{2n \ln(2n)}$, (2) $|z_n(A, \boldsymbol{x})| \le \sqrt{2n \ln(2n)}$, (3) $\bar{R}^n(A, \boldsymbol{x}) \ge 1/4 + \delta - O(1/n)$.*

Lemma 3. *Let δ be any positive constant and A be an algorithm such that $\bar{R}^n(A, D) \ge 1/4 + \delta$. Then S^δ_A is not empty.*

Proof: Since y_n and z_n are martingale sequences, we can apply Azuma's inequality to show that $\Pr[y_n \ge \sqrt{2n \ln(2n)}] < 1/n$ and $\Pr[z_n \ge \sqrt{2n \ln(2n)}] < 1/n$. Thus, since rewards are bounded by a constant value in our construction (namely 2), the contribution of sequences for which y_n or z_n are larger than $\sqrt{2n \ln(2n)}$ to the expected average reward is bounded by $O(1/n)$. This implies that if there exists an algorithm A such that $\bar{R}^n(A, D) \ge 1/4 + \delta$, then there exists a sequence $\boldsymbol{x}$ for which the $\bar{R}^n(A, \boldsymbol{x}) \ge 1/4 + \delta - O(1/n)$ and both y_n and z_n are bounded by $\sqrt{2n \ln(2n)}$. □

Now we would like to analyze the performance of an algorithm for some sequence $\boldsymbol{x}$ in S^δ_A. We first analyze the balanced sequence where $y_n = 0$ with a balanced algorithm (so $z_n = 0$), and then show how the analysis easily extends to sequences in the set S_A. In particular, we will first show that for the balanced sequence the optimal policy in terms of the objective function achieved has one fixed policy in times $[1, n]$ and another fixed policy in times $[n + 1, 2n]$. Due to lack of space the proof, which is similar but slightly more complicated than the proof of Lemma 1, is omitted.

Lemma 4. *Let $\boldsymbol{x} \in S$ be a sequence with $y_n = 0$ and let $A^{\boldsymbol{x}}_0$ be the set of algorithms for which $z_n = 0$ on $\boldsymbol{x}$. Then the optimal algorithm in $A^{\boldsymbol{x}}_0$ with respect to the objective function $MV(A, \boldsymbol{x})$ has a fixed policy in times $[1, n]$ and a fixed policy in times $[n + 1, 2n]$.*

Now that we have characterized the optimal algorithm for the balanced setting, we will analyze its performance. The next lemma connects the average reward to the standard deviation on balanced sequences by using the fact that on balanced sequences algorithms behave as they are "expected." The proof is again omitted due to lack of space.

Lemma 5. *Let $\boldsymbol{x} \in S$ be a sequence with $y_n = 0$, and let $A_0^{\boldsymbol{x}}$ be the set of algorithms with $z_n = 0$ on $\boldsymbol{x}$. For any positive constant δ, if $A \in A_0^{\boldsymbol{x}}$ and $\bar{R}^n(A, \boldsymbol{x}) = 1/4 + \delta$, then $\sigma^n(A, \boldsymbol{x}) \geq \frac{4\delta}{3}$.*

We now provide a bound on the objective function at time $2n$ given its average reward at time n. The proof uses the simple fact the added standard deviation is at least as large as the added average reward and thus cancels it. Once again, the proof is omitted due to lack of space.

Lemma 6. *Let $\boldsymbol{x}$ be any sequence and A any algorithm. If $\bar{R}^n(A, \boldsymbol{x}) = 1/4 + \delta$, then $MV^{2n}(A, \boldsymbol{x}) \leq 1/4 + \delta$ for any positive constant δ.*

Recall that the best expert at time n is expert 2 with reward $1/4$ and standard deviation 0, and the best expert at time $2n$ is expert 1 with average reward 1 and standard deviation $1/\sqrt{2}$. Using this knowledge in addition to Lemmas 5 and 6, we obtain the following proposition for the balanced sequence:

Proposition 1. *Let $\boldsymbol{x} \in S$ be a sequence with $y_n = 0$, and let $A_0^{\boldsymbol{x}}$ be the set of algorithms with $z_n = 0$ for s. If $A \in A_0^{\boldsymbol{x}}$, then A has a constant regret at either time n or time $2n$ or at both.*

We are now ready to return to the non-balanced setting in which y_n and z_n may take on values other than 0. Here we use the fact that there exists a sequence in S for which the average reward is at least $1/4 + \delta - O(1/n)$ and for which y_n and z_n are small. The next lemma shows that standard deviation of an algorithm A on sequences in S_A^δ is high at time n. The proof uses the fact that such sequences and algorithm can be changed with almost no effect on average reward and standard deviation to balanced sequence, for which we know the standard deviation of any algorithm must be high. The proof is omitted due to lack of space.

Lemma 7. *Let δ be any positive constant, A be any algorithm, and $\boldsymbol{x}$ be a sequence in S_A^δ. Then $\sigma^n(A, \boldsymbol{x}) \geq \frac{4\delta}{3} - O\left(\sqrt{\ln(n)/n}\right)$.*

We are ready to prove the main theorem of the section.

Proof: [Theorem 2] Let δ be any positive constant. If $\bar{R}^n(A, D) < 1/4 + \delta$, then there must be a sequence $\boldsymbol{x} \in S$ with $y_n \leq \sqrt{2n \ln(2n)}$ and $\bar{R}^n(A, \boldsymbol{x}) < 1/4 + \delta$. Then the regret of A at time $2n$ will be at least $1 - 1/\sqrt{2} - 1/4 - \delta - O(1/n)$.

If, on the other hand, $\bar{R}^n(A, D) \geq 1/4 + \delta$, then by Lemma 3 there exists a sequence $\boldsymbol{x} \in S$ such that $\bar{R}^n(A, \boldsymbol{x}) \geq 1/4 + \delta - O(1/n)$. By Lemma 7, $\sigma^n(A, \boldsymbol{x}) \geq 4/3\delta - O\left(\sqrt{\ln(n)/n}\right)$, and thus the algorithm has regret at time n of at least $\delta/3 - O\left(\sqrt{\ln(n)/n}\right)$. This shows that for any δ we have that either the regret at time n is constant or the regret at time $2n$ is constant. □

In fact we can extend this theorem to the broader class of objective functions of the form $\bar{R}^n(k, \boldsymbol{x}) - \alpha\sigma^n(A, \boldsymbol{x})$, where $\alpha > 0$ is constant. The proof is similar to the proof of Theorem 2 and the sequences used are built similarly. Both the constant and the length of the sequence will depend on α. The proof is omitted due to limits on space.

Theorem 3. *Let A be any online algorithm and α be a nonnegative constant. There exists a sequence $\boldsymbol{x}$ for which the regret of A with respect to the metric $\bar{R}^n(k, \boldsymbol{x}) - \alpha\sigma^n(A, \boldsymbol{x})$ is constant for some positive constant that depends on α.*

5 A Bicriteria Upper Bound

In this section we show that the recent algorithm of Cesa-Bianchi et al. [3] can yield a risk-reward balancing bound. Their original result expressed a no-regret bound with respect to *rewards* only, but the regret itself involved a variance term. Here we give an alternate analysis demonstrating that the algorithm actually respects a risk-reward trade-off. The quality of the results here depends on the bound M on the absolute value of expert rewards as we will show.

We first describe the Cesa-Bianchi et al. algorithm, **prod**(η). The algorithm has a parameter η and it maintains a set of K weights. The (unnormalized) weights $\tilde{w}_t^k$ are initialized to $\tilde{w}_t^k = 1$ for every expert k and updated according to $\tilde{w}_t^k \leftarrow \tilde{w}_{t-1}^k(1 + \eta x_{t-1}^k)$, where $\tilde{W}_t = \sum_{j=1}^k \tilde{w}_t^j$. The normalized weights at each time step are then defined as $w_t^k = \tilde{w}_t^k / \tilde{W}_t$.

Theorem 4. *For any expert $k \in \boldsymbol{K}$, for any $L \geq 2$, for the algorithm* **prod**(η) *with $\eta \geq 1/(LM)$ we have at time t*

$$\left(\frac{L\bar{R}^t(k, \boldsymbol{x})}{L+1} - \frac{\eta(3L+2)Var^t(k, \boldsymbol{x})}{6L}\right) - \frac{\ln K}{\eta} \leq \left(\frac{L\bar{R}^t(A, \boldsymbol{x})}{L-1} - \frac{\eta(3L-2)Var^t(A, \boldsymbol{x})}{6L}\right)$$

for any reward sequence $\boldsymbol{x}$ in which the absolute value of each reward is bounded by M.

The two expressions in parentheses in Theorem 4 both additively balance rewards and variance of rewards, but with differing coefficients. It is tempting but apparently not possible to convert this inequality into a competitive ratio. Nevertheless, as we now show, certain natural settings of the parameters cause the two expressions to give quantitatively similar trade-offs.

Let $\boldsymbol{x}$ be any sequence of rewards which are bounded in $[-1, 1]$, and let A be **prod**(η) for $\eta = 1/9$. Then for any time t and expert k we have

$$\left(0.9\bar{R}^t(k, \boldsymbol{x}) - 0.06Var^t(k, \boldsymbol{x})\right) - (9\ln K)/t \leq \left(1.125\bar{R}^t(A, \boldsymbol{x}) - 0.051Var^t(A, \boldsymbol{x})\right)$$

While the two trade-offs in this setting of the parameters are quite similar, the rewards coefficient is an order of magnitude larger than the variance coefficient in both. Now suppose $\boldsymbol{x}$ contains rewards bounded by a narrower bound $[-.1, .1]$ Let A be $\mathbf{prod(}\eta\mathbf{)}$ for $\eta = 1$. Then for any time t and expert k we have

$$\big(0.91\bar{R}^t(k,\boldsymbol{x}) - 0.533Var^t(k,\boldsymbol{x})\big) - (10\ln K)/t \leq \big(1.11\bar{R}^t(A,\boldsymbol{x}) - 0.466Var^t(A,\boldsymbol{x})\big)$$

This gives a much more even balance between rewards and variance on both sides. We note that the choice of a "reasonable" bound on the rewards magnitudes should be related to the time scale of the process — for instance, returns on the order of ±10% might be entirely reasonable annually but not daily.

The following facts about the behavior of $\ln(1+z)$ for small values of z will be useful in the proof of Theorem 4.

Lemma 8. *For any $L > 2$ and any v, y, and z such that $|v|$, $|y|$, $|v+y|$, and $|z|$ are all bounded by $1/L$ we have the following*

$$z - \frac{(3L+2)z^2}{6L} < \ln(1+z) < z - \frac{(3L-2)z^2}{6L}$$

$$\ln(1+v) + \frac{Ly}{L+1} < \ln(1+v+y) < \ln(1+v) + \frac{Ly}{L-1}$$

Similar to the analysis in [3], we bound $\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1}$ from above and below to prove Theorem 4. We start by bounding it from above.

Lemma 9. *For the algorithm* $\mathbf{prod(}\eta\mathbf{)}$ *with $\eta = 1/LM \leq 1/4$ we have,*

$$\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1} \leq \frac{\eta L R^n(A,\boldsymbol{x})}{L-1} - \frac{\eta^2(3L-2)nVar^n(A,\boldsymbol{x})}{6L}$$

at any time n for sequence $\boldsymbol{x}$ with the absolute value of rewards bounded by M.

Proof: Similarly to [3] we obtain,

$$\begin{aligned}\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1} &= \sum_{t=1}^{n}\ln\frac{\tilde{W}_{t+1}}{\tilde{W}_t} = \sum_{t=1}^{n}\ln\left(\sum_{k=1}^{K}\frac{\tilde{w}_t^k}{\tilde{W}_t}(1+\eta x_t^k)\right) = \sum_{t=1}^{n}\ln(1+\eta x_t^A)\\ &= \sum_{t=1}^{n}\ln(1+\eta(x_t^A - \bar{R}^n(A,\boldsymbol{x}) + \bar{R}^n(A,\boldsymbol{x})))\end{aligned}$$

Now using Lemma 8 twice we obtain the proof. □

Next we bound $\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1}$ from below. The proof is based on similar arguments to the previous lemma and the observation made in [3] that $\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1} \geq \ln\left(\frac{\tilde{w}_{n+1}^k}{K}\right)$, and is thus omitted.

Lemma 10. *For the algorithm* $\mathbf{prod(}\eta\mathbf{)}$ *with $\eta = 1/LM$ where $L \geq 2$, for any expert $k \in \boldsymbol{K}$ the following is satisfied*

$$\ln\frac{\tilde{W}_{n+1}}{\tilde{W}_1} \geq -\ln K + \frac{\eta L R^n(k,\boldsymbol{x})}{L+1} - \frac{\eta^2(3L+2)nVar^n(k,\boldsymbol{x})}{6L}$$

at any time n for any sequence $\boldsymbol{x}$ with rewards absolute values bounded by M.

Combining the two lemmas we obtain Theorem 4.

6 No-Regret Results for Localized Risk

In this section we show a no-regret result for an algorithm optimizing an alternative objective function that incorporates both risk and reward. The primary leverage of this alternative objective is that risk is now measured only "locally" — thus, the goal is to balance immediate rewards on the one hand with how far these immediate rewards deviate from the average rewards over some "recent" past on the other hand. In addition to allowing us to skirt the strong impossibility results for no-regret in the standard Sharpe and MV measures, we note that our new objective may be of independent interest, as it incorporates certain other notions of risk that are commonly considered in finance, where short-term volatility is usually of greater concern than long-term. For example, our new objective has the flavor of what is sometimes called "maximum draw-down", which is the largest decline in the price of a stock over a given, usually short, time period.

Consider the following measure of risk for an expert $k \in \mathbf{K}$ on a sequence of expert rewards $\boldsymbol{x}$:

$$P(k, \boldsymbol{x}) = \sum_{t=2}^{n} (x_t^k - \mathrm{AVG}_\ell^*(x_1^k, ..., x_t^k))^2$$

where $\mathrm{AVG}_\ell^*(x_1^k, .., x_n^k) = \sum_{t=0}^{\ell-1} (x_{n-t}^k/\ell)$ is the fixed window size average for some window size $\ell > 0$. [3]

The new risk-sensitive criterion will be $G^n(A, \boldsymbol{x}) = \bar{R}^n(A, \boldsymbol{x}) - \frac{P(A,\boldsymbol{x})}{n}$.

Our first observation is that the measure of risk defined here can be very similar to variance. In particular, if we let for every expert $k \in \mathbf{K}$, $p_t^k = (x_t^k - \mathrm{AVG}_t^*(x_1^k, .., x_t^k))^2$, then

$$\frac{P^n(k, \boldsymbol{x})}{n} = \frac{\sum_{t=2}^{n} p_t^k}{n} \ ; \ Var^n(k, \boldsymbol{x}) = \frac{\sum_{t=2}^{n} p_t^k (1 + \frac{1}{t-1})}{n}$$

Note that our measure differs from the variance in two aspects. The first is that in standard measures like variance, the variance of the sequence will be affected by rewards in the past and the future, whereas our measure depends only on rewards in the past. The second is the window size where the current reward is compared only to the rewards in the recent past, and not to all past rewards. While both of these differences are exploited in the proof, the fixed window size plays the more central role.

The main obstacle of the adaptive algorithms in the previous sections was the "memory" of the variance, which prevented them switching between the experts. The memory of the penalty now is ℓ and indeed our results will be meaningful when $\ell = o(\sqrt{T})$.

[3] Instead of taking fixed window size we could have taken the moving average, i.e.$\mathrm{AVG}^*(x_1, .., x_n) = (1 - \gamma) \sum_{t=1}^{n} \gamma^{n-t+1} x_t$ all results would apply for it (for an appropriate choice of γ)

The algorithm we discuss will work by feeding modified instantaneous gains to any best experts algorithm that satisfies the assumption below. This assumption is met by algorithms such as the weighted majority [5, 2] and EG [4].

Definition 2. *An optimized best expert algorithm is an algorithm that guarantees that for any sequence of reward vectors $\boldsymbol{x}$ over experts $\boldsymbol{K} = \{1, \ldots, K\}$, the algorithm selects a distribution w_t over $\boldsymbol{K}$ (using only the previous reward functions) such that*

$$\sum_{t=1}^{T}\sum_{k=1}^{K} w_t^k x_t^k \geq \sum_{t=1}^{T} x_t^k - \sqrt{TM \log K},$$

where $|x_t^k| \leq M$ and k is any expert. Furthermore, we also assume that decision distributions do not change quickly: $\|\boldsymbol{w_t} - \boldsymbol{w_{t+1}}\|_1 \leq \sqrt{\log(K)/t}$.

Since the risk function now has shorter memory, there is hope that a standard best expert algorithms will work. Therefore, we would like to incorporate this risk term into the instantaneous rewards fed to the best experts algorithm. We will define this instantaneous quantity, the *gain* of expert k at time t to be $g_t^k = x_t^k - (x_t^k - AVG^*(x_1^k, ..., x_{t-1}^k))^2 = x_t^k - p_t^k$, where p_t^k is the *penalty* for expert k at time t. It is natural to wonder whether $p_t^A = \sum_{k=1}^{K} w_t^k p_t^k$; unfortunately, this is not the case. Fortunately, we can show that they are similar. To formalize the connection between the measures, we let $\hat{P}(A, \boldsymbol{x}) = \sum_{t=1}^{T}\sum_{k=1}^{K} w_t^k p_t^k$ be the weighted penalty function of the experts, and $P(A, \boldsymbol{x}) = \sum_{t=1}^{T} p_t^A$ be the penalty function observed by the algorithm. The next lemma relates these quantities.

Lemma 11. *Let $\boldsymbol{x}$ be any reward sequence such that all rewards are bounded by M. Then $\hat{P}^T(A, \boldsymbol{x}) \geq P^T(A, \boldsymbol{x}) - O\left(TM^2\ell\sqrt{\frac{\log K}{T-\ell}}\right)$.*

Proof:

$$\begin{aligned}
\hat{P}^T(A, \boldsymbol{x}) &= \sum_{t=1}^{T}\sum_{k=1}^{K} w_t^k (x_t^k - AVG_\ell^*(x_1^k, .., x_t^k))^2 \geq \sum_{t=1}^{T}\left(\sum_{k=1}^{K} w_t^k \left(x_t^k - \frac{\sum_{j=1}^{\ell} x_{t-j+1}^k}{\ell}\right)\right)^2 \\
&= \sum_{t=1}^{T}\left(\sum_{k=1}^{K} w_t^k x_t^k - \frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell}(w_t^k - w_{t-j+1}^k + w_{t-j+1}^k)x_{t-j+1}^k}{\ell}\right)^2 \\
&= \sum_{t=1}^{T}\left(\left(\sum_{k=1}^{K} w_t^k x_t^k - \frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell} w_{t-j+1}^k x_{t-j+1}^k}{\ell}\right)^2 + \left(\frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell} \epsilon_j^k x_{t-j+1}^k}{\ell}\right)^2\right. \\
&\qquad \left. -2\left(\frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell} \epsilon_j^k x_{t-j+1}^k}{\ell}\right)\left(\sum_{k=1}^{K} w_t^k x_t^k - \frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell} w_{t-j+1}^k x_{t-j+1}^k}{\ell}\right)\right) \\
&\geq P^T(A, \boldsymbol{x}) - \sum_{t=1}^{T}\left(2M\frac{\sum_{k=1}^{K}\sum_{j=1}^{\ell} |\epsilon_j^k| M}{\ell}\right) \\
&\geq P^T(A, \boldsymbol{x}) - 2M^2\ell T\sqrt{\frac{\log K}{T-\ell}} \geq P^T(A, \boldsymbol{x}) - O\left(TM^2\ell\sqrt{\frac{\log K}{T-\ell}}\right)
\end{aligned}$$

where $\epsilon_j^k = w_t^k - w_{t-j+1}^k$. The first inequality is an application of Jensen's inequality using the convexity of x^2. The third inequality follows from the fact that $\sum_{k=1}^K |\epsilon_j^k|$ is bounded by $j\sqrt{\frac{\log K}{T-j}}$ using our best expert assumption. □

Next we we state the main result of this section which is a no-regret algorithm with the risk-sensitive function G.

Theorem 5. *Let A be a best expert algorithm that satisfies Definition 2 with instantaneous gain function $g_t^k = x_t^k - (x_t^k - AVG^*(x_1^k, ..., x_{t-1}^k))^2$ for expert k at time t. Then for large enough T for any reward sequence $\boldsymbol{x}$ and any expert k we have for window size ℓ*

$$G(k, \boldsymbol{x}) - O\left(M^2 \ell \sqrt{\frac{\log K}{T - \ell}}\right) \leq G(A, \boldsymbol{x})$$

Proof:

$$\begin{aligned}
T \cdot G(k, \boldsymbol{x}) &= \sum_{t=1}^{T} x_t^k - \sum_{t=1}^{T} (x_t^k - AVG_\ell^*(x_1^k, .., y_t^k))^2 \\
&\leq \sum_{t=1}^{T} \sum_{k'=1}^{K} w_t^{k'} x_t^{k'} - \sum_{t=1}^{T} \sum_{k'=1}^{K} w_t^{k'} (x_t^{k'} - AVG_\ell^*(x_1^{k'}, .., x_t^{k'}))^2 + \sqrt{TM \log K} \\
&\leq T \cdot G(A, \boldsymbol{x}) + O\left(TM^2 \ell \sqrt{\frac{\log K}{T - \ell}}\right) + \sqrt{TM \log K}
\end{aligned}$$

The first inequality is due to the best expert algorithm, and the last inequality is due to Lemma 11. □

Corollary 1. *Let A be a best expert algorithm that satisfies Definition 2 with instantaneous reward function $g_t^k = x_t^k - (x_t^k - AVG^*(x_1^k, ..., x_{t-1}^k))^2$. Then for large enough T we have for any expert k and fixed window size $\ell = O(\log T)$*

$$G(k, \boldsymbol{x}) - \tilde{O}\left(M^2 \sqrt{\frac{\log K}{T}}\right) \leq G(A, \boldsymbol{x})$$

7 Simulations

We conclude by briefly showing the results of some preliminary simulations on the algorithms and measures discussed. Despite the fact that neither of the algorithms given are provably competitive with the Sharpe and MV measures, we examine their performance on these standards in comparison to EG. The left panel of Figure 1 shows the price time series for $K = 2$ simulated stocks. These time series were generated from a stochastic model that divides 10000 steps into blocks of size 100. Within each block one of the two stocks is generally trending up, while the other is trending down, with the choice of which stock is trending

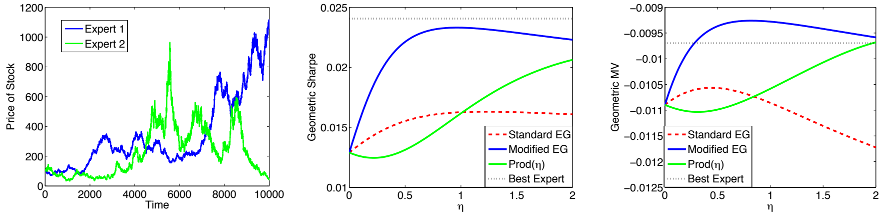

Fig. 1. Left: The price time series of two experts. **Center:** The geometric Sharpe value achieved by each algorithm. **Right:** The geometric MV achieved by each algorithm.

up made randomly (details omitted). This is one particular model that generates data for which standard algorithms like EG with small η outperform uniform constant rebalanced ($\eta = 0$), so the learning helps[4].

The center and right panels compare the three algorithms — standard (risk-insensitive) EG, our modified version of EG with window size $\ell = \sqrt{T} = 100$, and $\mathbf{prod}(\eta)$ as a function of η on both Sharpe ratio (center panel) and MV (right panel). The performance of the best expert with respect to each measure is also shown. Note that both of the algorithms that take risk into account perform noticeably better than standard EG on both risk-reward measures. In particular, our modified version of the EG actually beats the best expert in MV when run with moderately small values of η.

These simulations are still preliminary; we expect to expand them in upcoming work.

References

1. Zwi Bodie, Alex Kane, and Alan J. Marcus. Portfolio Performance Evaluation, Investments, 4th edition,Irwin McGraw-Hill, 1999.
2. N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R.E. Schapire, and M.K. Warmuth. How to Use Expert Advice, *J. of the ACM*, Vol 44(3): 427-485, 1997.
3. N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved Second-Order Bounds for Prediction with Expert Advice, *COLT*, 217–232, 2005.
4. D.P. Helmbold, R.E. Schapire, Y. Singer, and M.K. Warmuth. On-line portfolio selection using multiplicative updates, *Mathematical Finance*, 8(4), 325–347, 1998.
5. Nick Littlestone and Manfred K. Warmuth. The Weighted Majority Algorithm, *Information and Computation,* 108(2): 212-261, 1994.
6. Harry Markowitz. Portfolio Selection, *The Journal of Finance*, 7(1):77–91, 1952.
7. William F. Sharpe. Mutual Fund Performance, *The Journal of Business*, Vol 39, Number 1, part 2: Supplement on Security Prices, 119-138, 1966.

[4] In contrast, running EG at small learning rates on the last 6 years of S&P 500 closing price data *underperforms* uniform rebalanced despite the theoretical guarantees.