Optimal amortized regret in every interval

Rina Panigrahy
Microsoft Research
Mountain View, CA
rina@microsoft.com

Preyas Popat *
Google Inc.
Mountain View, CA
preyas.p@gmail.com

Abstract

Consider the classical problem of predicting the next bit in a sequence of bits. A standard performance measure is *regret* (loss in payoff) with respect to a set of experts. For example if we measure performance with respect to two constant experts one that always predicts 0's and another that always predicts 1's it is well known that one can get regret $O(\sqrt{T})$ with respect to the best expert by using, say, the weighted majority algorithm [LW89]. But this algorithm does not provide performance guarantee in any interval. There are other algorithms (see [BM07, FSSW97, Vov99]) that ensure regret $O(\sqrt{x \log T})$ in any interval of length x. In this paper we show a randomized algorithm that in an amortized sense gets a regret of $O(\sqrt{x})$ for any interval when the sequence is partitioned into intervals arbitrarily. We empirically estimated the constant in the $O()$ for T upto 2000 and found it to be small – around 2.1. We also experimentally evaluate the efficacy of this algorithm in predicting high frequency stock data.

1 INTRODUCTION

Consider the following classical game of predicting a binary ± 1 sequence. An algorithm A sees a binary sequence $\{b_t\}_{t\geq 1}$, one bit at a time, and attempts to predict the next bit b_t from the past history $b_1, \dots b_{t-1}$. The *payoff* A_T of the algorithm in T steps is the number of correct guesses minus the number of the wrong guesses. In other words, let $\tilde{b}_t \in [-1, 1]$ be the prediction for the t^{th} bit based on the previous bits then:

$$A_T := \sum_{1 \leq t \leq T} b_t \tilde{b}_t.$$

The payoff per time step $b_t \tilde{b}_t$ is essentially equivalent to the well known absolute loss function $|b_t - \tilde{b}_t|$ (see for example [CBL06], chapter 8).[1]

One can view this game as an idealized "stock prediction" problem as follows. In each unit time, the stock price goes up or down by precisely \$1, and the algorithm bets on this event. If the bet is right, the player wins one dollar, and otherwise loses one dollar. Not surprisingly, in general, it is impossible to guarantee a positive payoff for all possible scenarios (sequences). However, one could hope to give some guarantees on the payoff of the algorithm based on certain properties of the sequence.

For example one can compare the payoff to the better of two choices (experts), which correspond to two constant algorithms: first one, where $\tilde{b}_t = +1$ and the second one where $\tilde{b}_t = -1$ for all t. Note that the best of these experts gets payoff $|\sum_{1\leq t \leq T} b_t|$, which corresponds to the "optimal in hindsight" expert among the two choices. The *regret* of an algorithm is defined as how much worse the algorithm performs as opposed to the best of the two experts (in hindsight, after seeing the sequence). This has been studied in a number of papers, including

*This work was done while this author was at Microsoft Research.

[1] since when $|b_t| = 1$, $|b_t - \tilde{b}_t| = |b_t||b_t - \tilde{b}_t| = |1 - b_t\tilde{b}_t| = 1 - b_t\tilde{b}_t$. Thus the absolute loss function is the negative of our payoff in one step plus a shift of 1. Also b_t values from $\{-1, 1\}$ or $\{0, 1\}$ are equivalent by a simple scaling and shifting transform.

[Cov65, LW89, Cov91, ACBFS02, AB09]. A classical result says that one can obtain a regret of $\Theta(\sqrt{T})$ for a sequence of length T, via, say, the weighted majority algorithm [LW89]. Formally, for a sequence $X = b_1, \ldots, b_T$, let $h(X) = \sum_{1 \le t \le T} b_t$ denote the "height" of the sequence when plotted cumulatively as a chart. Then we have the following theorem:

Theorem 1.1 *[Cov65, CBFH+97] There is an algorithm that achieves payoff $\ge |h(X)| - \alpha\sqrt{T}$. It is also known that the optimal value of $\alpha \to \sqrt{2/\pi}$ as $T \to \infty$.*

However, an algorithm that only focuses on the overall regret does not exploit short term trends in the sequence and only relies on a 'global' long term bias in the full string. Consider for example a sequence that may not have a high overall bias but has many intervals in which there may be a high level of bias. Our result is that for any partitioning of the sequence into intervals, one can essentially get a regret proportional to $\sqrt{x}$ for each interval of length x in an amortized sense (Theorem 1.3). Although our results are stated for bits they work even when b_t is a real number in $[-1, 1]$. We note that even though similar bounds have been obtained before ([BM07, FSSW97, Vov99] and, more recently, [HS09, KP11]), the penalty on an interval of length x is $O(\sqrt{x \log T})$ in these previous results. Note that in adversarial settings one is interested in a prediction algorithm that can get a positive payoff even if the sequence departs slightly from random; or we may ask what is the smallest amount non-randomness that can be "noticed" by the prediction algorithm. So while our result may seem like just shaving a $\log T$ factor, the reason $\sqrt{x}$ is much better than $\sqrt{x\ logT}$ is that in certain adversarial settings (like financial markets), the uptrend or downtrend in total per interval may not be too far from that of a random sequence. Note that a random ± 1 sequence of length x has a height of magnitude $\Theta(\sqrt{x})$ in expectation. So we are saying that even if the height of a sequence of length x is some constant multiple of $\sqrt{x}$, we get a positive payoff.

The bit prediction problem we consider is closely related to the two experts problem (or multi-armed bandits problem with full information). In each round each expert has a payoff in the range $[0, 1]$ that is unknown to the algorithm. For two experts, let b_{1t}, b_{2t} denote the payoffs of the two experts at time t. The algorithm pulls each arm (expert) with probability $\tilde{b}_{1t}, \tilde{b}_{2t} \in [0, 1]$ respectively where $\tilde{b}_{1t} + \tilde{b}_{2t} = 1$. The (expected) payoff of the algorithm in this setting is $A'_T := \sum_{t=1}^{T} b_{1t}\tilde{b}_{1t} + b_{2t}\tilde{b}_{2t}$.

We will be concerned with the following payoff function in this paper:

Definition 1.2 *(Interval payoff function: P_α)*

Let $X_1, \ldots, X_k$ denote a partition of the sequence X into a disjoint union of k intervals, that is, X is the concatenation of these k subsequences. We will use $h(X_i)$ to denote the sum of the bits in the interval X_i and $|X_i|$ to denote the length of X_i.

The interval payoff function, $P_\alpha(X)$ is defined as the maximum value of the expression

$$\sum_{i=1}^{k} \left(|h(X_i)| - \alpha\sqrt{|X_i|}\right)$$

over all $1 \le k \le |X|$ and all partitions $X_1, \ldots, X_k$ of X.

We say that a payoff function $f : \{-1, 1\}^T \to \mathbb{R}$ is feasible if there is a bit prediction algorithm which on sequence X achieves payoff at least $f(X)$.

Theorem 1.3 **(Main Theorem)** *There is an absolute constant $0 < \alpha < 10$ independent of T such that the interval payoff function P_α is feasible.*

For the two experts problem our result translates to the following guarantee:

$$A'_T \ge \sum_{i=1}^{k} \left(\max_{j \in 1,2} \left(\sum_{t \in X_i} b_{jt} \right) - \frac{\alpha}{2}\sqrt{|X_i|} \right).$$

Here $\sum_{t \in X_i} b_{jt}$ is the payoff of the j^{th} expert in the interval X_i.

This result can be viewed as incurring a penalty of $\alpha\sqrt{|X_i|}$ for each interval X_i. We theoretically show that the optimal value of α is at most 10 (Section 2). We empirically estimated the optimal α for T up to 2000 and found it to be small – around 2.1 (Section 4.1).

We stress here that the algorithm doesn't need to know the partition or the length of the partition in advance. We also note that our guarantee does not hold for each interval individually but when we look at the net payoff in an amortized sense, we may account for a regret of at most $\alpha\sqrt{|X|}$ for an interval of length X. In fact, the guarantee is impossible to achieve in a non-amortized sense. We show that if we measure regret based on the performance of an algorithm in a given interval then one will have to trade-off regrets at different time scales. The following observation is proven in the full version [PP13].

Observation 1.4 *There is no prediction algorithm that can guarantee a regret of $O(\sqrt{|Y|})$ on all intervals Y for all input sequences.*

Regarding the computation of P_α, we show:

Theorem 1.5 *The value of $P_\alpha(S)$ for a particular sequence S of length T can be computed using dynamic programming in time $O(T^3)$.*

For a given T, let $\alpha_0(T)$ denote the minimum α such that P_α is feasible for all sequences of length T. It is possible to determine α_0 using the following well known observation by Cover.

Observation 1.6 (Cover [Cov65]) *A payoff function $f : \{-1,1\}^T \rightarrow \mathbb{R}$ is feasible if and only if $E_S[f(S)] \leq 0$ where S is a uniformly random sequence in $\{-1,1\}^T$.*

This is achieved by a prediction algorithm that predicts $\tilde{b}_t = \frac{E_U[f(s.1.U)]-E_U[f(s.(-1).U)]}{2}$ where s is the sequence of bits seen so far, U is a suffix sequence chosen uniformly at random and $s.b.U$ denotes the concatenated sequence starting with s followed by bit b followed by the sequence U. Note that $\tilde{b}_t \in [-1,1]$ as long as for all s, $|E_U[f(s.1.U)] - E_U[f(s.(-1).U)]| \leq 2$

Algorithm and Running time: Theorem 1.5 and Observation 1.6 suggest a simple algorithm for achieving payoff function P_α. Take the sequence s seen so far, append a $+1$ and then a random sequence to make it into a complete sequence of length T. Compute $P_\alpha(S)$ for the resulting sequence S. Do this again replacing the $+1$ by a -1. Predict $\tilde{b}_t$ to be the half of the difference in the two cases.

We note that a deterministic algorithm achieving the guarantee of Theorem 1.3 may take exponential time since it would need to find $P_\alpha(S)$ for every random completion of the bits seen so far. Alternatively, there is a simple randomized algorithm which achieves the same payoff in expectation by taking a different random completion for every prefix. A naive implementation of this randomized algorithm will take T^3 time for each bit being predicted. We show a simple variant that reduces this to $O(\log T)$ time with precomputation.

Theorem 1.7 *There is a randomized algorithm that achieves the payoff guarantee P_α of Theorem 1.3 in expectation and spends $O(T^2)$ time per step. There is also a randomized algorithm that achieves payoff $P_{\alpha'}$ with $\alpha' = c\alpha$ and spends only $O(\log T)$ time per step. Here $c := \frac{\sqrt{2}}{\sqrt{2}-1}$.*

These algorithm use pre-computed information that takes time $O(T^2)$ and $O(T \log T)$ time to compute for the first and the second algorithm respectively.

Generalization to real numbers: In the full version [PP13], we show that a variant of the guarantee holds in a semi-adversarial model where a string of real numbers may be chosen instead of bits. The model combines worst case and average case settings where the signs of the real numbers may be chosen adversarially (that is, in the worst case) but the magnitudes of the real numbers come from a pre-specified distribution independently and randomly.

Experimental results: We implement our algorithm, the weighted majority algorithm, an algorithm based on Autoregressive Integrated Moving Average (ARIMA) and an algorithm of [KP11], and compare their performance when predicting financial time series data. Specifically, we consider the high frequency price data of 5 stocks, and we apply these algorithms to predict the per minute price changes in an online fashion taking the values in each day as a separate sequence. That is we predict the next minute returns of mid-prices for each stock based on its previous 1 minute returns in the day. We perform this experiment over 189 trading days for each stock and find that on an average our algorithm performs better than other prediction algorithms based on regret minimization but is outperformed by the ARIMA algorithm. On the other hand, as we discussed above, our algorithm has certain provable guarantees for *every* sequence which the ARIMA algorithm lacks. The experimental setup and results are described in more detail in Section 4.

1.1 Related work

There is large body on work on regret style analysis for prediction. Numerous works including [Cov65, CBFH+97] have examined the optimal amount of regret achievable with respect to two or more experts. A good reference for the results in this area is [CBL06]. It is well known that in the case of static experts, the optimal regret achievable is exactly equal to the Rademacher complexity of the predictions of the experts (chapter 8 in [CBL06]). Recent works such as [ALW06, AWY08, MS08] have extended this analysis to other settings. Measures other than the standard regret measure have been studied in [RST10]. The question of what can be achieved if one would like to have a significantly better guarantee with respect to a fixed expert or a distribution of experts was asked before in [EDKMW08, KP11]. Tradeoffs between regret and minimum payoff were also examined in [Vov98], where the author studied the set of values of a, b for which an algorithm can have payoff $aOPT + b \log N$, where OPT is the payoff of the best arm and a, b are constants.

Regret minimization algorithms with performance guarantees within each interval have been studied in [BM07, FSSW97, Vov99] and more recently in [HS09, KP11]. As we mentioned, some of these algorithms achieve a regret of $O(\sqrt{x \log T})$ for every interval of size x in a sequence of length T. A related work which also seeks to exploit short term trends in the sequence is [HW98], where the regret bound proportional to $\sqrt{Tk}$ in the best case where k is the number of intervals (see [CBL06], Corollary 5.1). The main difference between the work of [HW98] and our results is that their algorithm requires fixing the number of intervals, k, in advance whereas our algorithm works simultaneously for all k. Also note that their regret guarantee is always higher than the payoff function P_α for a sequence of length T achieving equality only in the special case when all intervals are of equal length T/k.

Numerous papers (for example [Blu97, HSSW98, AHKS06]) have implemented algorithms inspired from regret style analysis and applied it on financial and other types of data.

1.2 Overview of the proof

In this section we give a high level idea of our proof, the formal proof appears in Section 2.

To prove the main theorem we want to compute the minimum α such that $E_S[P_\alpha(S)] \leq 0$ (See Observation 1.6). We first introduce a variant of the payoff function $P_\alpha(S)$ as follows. Instead of computing the maximum value of $\sum_i |h(X_i)| - \alpha\sqrt{|X_i|}$ over all possible partitions, will only allow partitions where the intervals are of the form $(2^i j, 2^i(j+1)]$; that is, intervals that are obtained by dividing the string into segments of length that are some power of 2. We will refer to such intervals as 'aligned' intervals (Definition 2.3). Further we will only look at T values that is some power of 2. Note that any interval can be broken into at most $\log T$ aligned intervals. Let $P_\alpha^A(S)$ denote the maximum value of $\sum_i |h(X_i)| - \alpha\sqrt{|X_i|}$ with partitions into aligned intervals. We first show that

Lemma 1.8 *If $E[P_\alpha^A(S)] \leq 0$ then $E[P_{c\alpha}(S)] \leq 0$ where $c := \frac{\sqrt{2}}{\sqrt{2}-1}$.*

Next we show

Theorem 1.9 *There is an absolute constant $\alpha \leq 2.8$ such that $E[P_\alpha^A(S)] \leq 0$.*

We prove Theorem 1.9 recursively for T that are increasing powers of 2. We inductively show that the distribution of $P_\alpha^A(S)$ is stochastically upper bounded by a shifted exponential distribution (Definition 2.4) with certain parameters (Equation 2.1), where S is a uniformly random sequence of length T. Since we are dealing with splits into aligned intervals, we can assume that either the best split for S is the whole interval, or the mid-point of S is one of the splitting points. For the first case, we may upper bound the payoff function using Hoeffding's bound (Theorem 2.2), while for the second case we may inductively assume that the distribution of payoffs for the subsequences is stochastically bounded by a shifted exponential distribution. We then separately bound each of these distributions by the shifted exponential distribution.

2 FEASIBILITY OF PAYOFF FUNCTION P_α

2.1 Preliminaries

Definition 2.1 *(Binomial distribution B_n) Let $x_1, x_2, \ldots, x_n \in \{-1, 1\}$ be uniformly and independently distributed. Then the sum*

$$Y := \sum_{i=1}^{n} x_i$$

is said to be binomially distributed. We denote the distribution as B_n.

Theorem 2.2 *(Hoeffding's bound) [Hoe63]*

$$\Pr[|B_n| \geq y \cdot \sqrt{n}] \leq 2 \cdot \exp\left(-\frac{y^2}{2}\right)$$

Definition 2.3 *(Aligned interval)*

We assume here that T is a power of 2. An aligned interval is one which is obtained by breaking $[1, T]$ into 2^i equal parts for $i \in [0, \log T]$ and picking one of the parts. So for instance the first part is always $[1, T/2^i]$; each aligned interval can be written as $[jT/2^i + 1, (j+1)T/2^i]$ for non negative integers i and j.

We denote the interval payoff function corresponding to Definition 1.2 which allows only aligned splits as P_α^A.

Definition 2.4 *(Shifted Exponential distribution) The probability density function $f_{\mu,\sigma,n}$ of shifted exponential distribution with mean $\sigma\sqrt{n}$ and shift $\mu\sqrt{n}$ is defined as follows:*

$$f_{\mu,\sigma,n}(y) := \frac{1}{\sigma\sqrt{n}} \exp\left(-\frac{y - \mu\sqrt{n}}{\sigma\sqrt{n}}\right) \quad \forall y \geq \mu\sqrt{n}$$
$$f_{\mu,\sigma,n}(y) := 0 \quad \forall y \leq \mu\sqrt{n}$$

We denote a random variable distributed according to $f_{\mu,\sigma,n}$ as $F_{\mu,\sigma,n}$. That is, $\Pr[F_{\mu,\sigma,n} \geq y] = \int_y^\infty f_n(s)\,\mathrm{d}s = \exp\left(-\frac{y-\mu\sqrt{n}}{\sigma\sqrt{n}}\right)$ when $y \geq \mu\sqrt{n}$ and 1 otherwise.

2.2 Proof of feasibility

The following lemma is a restatement of lemma 1.8 and is proven using a standard doubling trick

Lemma 2.5 *If P_α^A is feasible then $P_{c\alpha}$ is also feasible, where $c := \frac{\sqrt{2}}{\sqrt{2}-1}$.*

Proof:

Let $X_1, X_2, \ldots, X_k$ denote a partition of a given sequence S. We split each interval X_i into a disjoint union of aligned intervals $Y_{i1}, \ldots, T_{il}$. We will then show that the identity

$$\sum_{j=1}^{l} \sqrt{|Y_{ij}|} \leq c \cdot \sqrt{|X_i|}$$

always holds where $|I|$ denotes the length of the interval I. This suffices to prove the theorem since $h(X_i) \leq \sum_{j=1}^{l} h(Y_{ij})$.

For notational simplicity, let $I = X_i$ and $x = |I|$. If I is an aligned interval we are done, otherwise we write it as the minimal union of aligned intervals (take out the largest aligned interval in I and repeat). There are three possibilities:-

1. $I = I_1 \cup I_2$ is a union of two intervals of size $x/2$ each (eg. the interval $[T/4+1,\ 3T/4]$)
2. $I = I_1 \cup I_2 \cup \ldots \cup I_l$, where each I_j is of a different size. Note that all interval sizes on the right are powers of 2 and strictly less than x
3. $I = J \cup J'$ where each J can be written as a union of intervals as in 1 or 2 above

In the first case,

$$\sqrt{|I_1|} + \sqrt{|I_2|} \leq 2 \cdot \sqrt{x/2} = \sqrt{2} \cdot \sqrt{x}$$

In the second case,

$$\sum_{j=1}^{l} \sqrt{|I_j|} \leq \sqrt{x} \cdot \sum_{j=1}^{\infty} \sqrt{1/2^j} = \frac{1}{\sqrt{2}-1} \cdot \sqrt{x}$$

In the third case,

$$\sqrt{|J|} + \sqrt{|J'|} \leq \frac{1}{\sqrt{2}-1} \cdot \sqrt{|J|} + \frac{1}{\sqrt{2}-1} \cdot \sqrt{|J'|} \leq \frac{\sqrt{2}}{\sqrt{2}-1} \cdot \sqrt{x}$$

■

We are now ready to prove Theorem 1.9.

Proof: [Proof of Theorem 1.9] We need to show that for all $T \geq 1$, $\mathbb{E}_{x \in \{-1,1\}^T}[P_\alpha^A(x)] \leq 0$. After that, the theorem follows from Observation 1.6 (it is easy to check that the condition required for required for $\tilde{b}_t \in [-1,1]$ given in Observation is satisfied by P_α^A).

We will prove the theorem by induction. We will show that when n is a power of 2,

$$\forall y \in \mathbb{R} \quad \Pr_{x \in \{-1,1\}^n}[P_\alpha^A(x) \geq y] \;\leq\; \Pr[F_{\mu,\sigma,n} \geq y] \tag{2.1}$$

for some $\mu := \mu(\alpha)$ and $\sigma := \sigma(\alpha)$. Here $F_{\mu,\sigma,n}$ is as in Definition 2.4.

Note that this would imply $\mathbb{E}_{x \in \{-1,1\}^n}[P_\alpha^A(x)] \leq \mathbb{E}[F_{\mu,\sigma,n}] = (\mu+\sigma)\sqrt{n}$. We will show that for a suitable choice of α, the term $\mu + \sigma \leq 0$, and this suffices to prove the theorem.

It remains to prove Equation 2.1. For the base case, $n = 1$, we see that the equation is satisfied for $\mu \geq 1 - \alpha$, $\sigma > 0$. We will now show that it is satisfied for $2n$ whenever it is satisfied for n (for appropriate μ and σ).

Now, for a sequence $x := (x_1, x_2) \in \{-1,1\}^n \times \{-1,1\}^n$, $P_\alpha^A(x) = \max(P_\alpha^A(x_1) + P_\alpha^A(x_2), |h(x)| - \alpha \cdot \sqrt{2n})$. So for every x such that $P_\alpha^A(x) \geq y$ we must have either $P_\alpha^A(x_1) + P_\alpha^A(x_2) \geq y$ or that $h(x) - \alpha \cdot \sqrt{2n} \geq y$. Thus,

$$\begin{aligned}
&\Pr_{x \in \{-1,1\}^{2n}}[P_\alpha^A(x) \geq y] && (2.2)\\
\leq &\Pr_{x_1,x_2 \in \{-1,1\}^n}[P_\alpha^A(x_1) + P_\alpha^A(x_2) \geq y] && (2.3)\\
&+ \Pr_{x \in \{-1,1\}^{2n}}[h(x) - \alpha \cdot \sqrt{2n} \geq y] && (2.4)\\
\leq &\Pr[F_{\mu,\sigma,n} + F'_{\mu,\sigma,n} \geq y] && (2.5)\\
&+ \Pr_{x \in \{-1,1\}^{2n}}[h(x) - \alpha \cdot \sqrt{2n} \geq y] && (2.6)
\end{aligned}$$

Here F and F' are independent random variables distributed as in Definition 2.4. We will show that the first and second term are each bounded by $\frac{1}{2}\Pr[F_{2n} \geq y]$ which is sufficient to prove Equation 2.1. Note that we only need to consider $y \geq \mu\sqrt{2n}$ since for smaller values of y we have

$$\Pr_{x \in \{-1,1\}^{2n}}[P_\alpha^A(x) \geq y] \leq \Pr[F_{2n} \geq y] = 1$$

Henceforth, we will use shorthands $f_n := f_{\mu,\sigma,n}$ and $F_n := F_{\mu,\sigma,n}$.

The first term can be written as:-

$$\begin{aligned}
\Pr[F_n + F'_n \geq y] &= \int_y^\infty \int_{-\infty}^\infty f_n(s) \cdot f_n(w-s)\, \mathrm{d}s\, \mathrm{d}w\\
&= \int_y^\infty \int_{\mu\sqrt{n}}^{w-\mu\sqrt{n}} f_n(s) \cdot f_n(w-s)\, \mathrm{d}s\, \mathrm{d}w
\end{aligned}$$

where the second equation follows from the fact that $f_n(s) = 0$ for $s < \mu\sqrt{n}$ and $f_n(w-s) = 0$ for

$s > w - \mu\sqrt{n}$. Thus, we need to show for all $y \geq \mu\sqrt{2n}$:-

$$\begin{aligned}
&\int_y^\infty \int_{\mu\sqrt{n}}^{w-\mu\sqrt{n}} f_n(s) \cdot f_n(w-s) \, \mathrm{d}s \, \mathrm{d}w \leq \frac{1}{2} \Pr[F_{2n} \geq y] \\
\Longleftarrow & \frac{1}{\sigma^2 n} \int_y^\infty \int_{\mu\sqrt{n}}^{w-\mu\sqrt{n}} \exp\left(\frac{-w+2\mu\sqrt{n}}{\sigma\sqrt{n}}\right) \, \mathrm{d}s \, \mathrm{d}w \\
& \leq \frac{1}{2} \exp\left(-\frac{y-\mu\sqrt{2n}}{\sigma\sqrt{2n}}\right) \\
\Longleftarrow & \frac{1}{\sigma^2 n} \int_y^\infty \int_{\mu\sqrt{n}}^{w-\mu\sqrt{n}} \exp\left(-\frac{w-2\mu\sqrt{n}}{\sigma\sqrt{n}}\right) \, \mathrm{d}s \, \mathrm{d}w \\
& \leq \frac{1}{2} \exp\left(-\frac{y-\mu\sqrt{2n}}{\sigma\sqrt{2n}}\right) \\
\Longleftarrow & \frac{1}{\sigma^2 n} \int_y^\infty (w-2\mu\sqrt{n}) \exp\left(-\frac{w-2\mu\sqrt{n}}{\sigma\sqrt{n}}\right) \, \mathrm{d}w \\
& \leq \frac{1}{2} \exp\left(-\frac{y-\mu\sqrt{2n}}{\sigma\sqrt{2n}}\right)
\end{aligned}$$

In the third line we implicitly assume that $y \geq 2\mu\sqrt{n}$, since otherwise the left hand side is less than 0 and the equation is satisfied.

Note that the integral is of the form $\int u \cdot e^{-cu}$ which integrates to $-\left(\frac{u+1/c}{c}\right) \cdot e^{-cu}$. Thus, integrating and substituting $z := y - 2\mu\sqrt{n}$ we need to show for all $z \geq 0$,

$$\begin{aligned}
& \frac{1}{\sigma\sqrt{n}} \cdot (z + \sigma\sqrt{n}) \cdot \exp\left(-\frac{z}{\sigma\sqrt{n}}\right) \\
& \leq \frac{1}{2} \exp\left(-\frac{z + (\sqrt{2}-1)\mu\sqrt{2n}}{\sigma\sqrt{2n}}\right) \\
\Longleftarrow & \frac{2z}{\sigma\sqrt{n}} + 2 \\
& \leq \exp\left(\frac{z}{\sigma\sqrt{n}} - \frac{z + (\sqrt{2}-1)\mu\sqrt{2n}}{\sigma\sqrt{2n}}\right) \\
\Longleftarrow & \frac{2z}{\sigma\sqrt{n}} + 2 \\
& \leq \exp\left(\frac{(\sqrt{2}-1)z}{\sigma\sqrt{2n}}\right) \cdot \exp\left((\sqrt{2}-1)\frac{-\mu}{\sigma}\right)
\end{aligned}$$

Substituting $w := \frac{z}{\sigma\sqrt{n}}$, we need for all $w \geq 0$,

$$\begin{aligned}
& 2w + 2 \\
& \leq \exp\left(\frac{(\sqrt{2}-1)w}{\sqrt{2}}\right) \cdot \exp\left((\sqrt{2}-1)\frac{-\mu}{\sigma}\right) \\
\Longleftarrow & \frac{2w+2}{\exp\left(\frac{(\sqrt{2}-1)w}{\sqrt{2}}\right)} \leq \exp\left((\sqrt{2}-1)\frac{-\mu}{\sigma}\right)
\end{aligned}$$

The left hand side is maximized at $w = 1/\sqrt{2}$ and the value of left hand side at that point is around 2.78. Thus, if $(-\mu/\sigma) \geq 2.47$ then the equation is always satisfied.

We now turn to bounding the second term 2.6. We need to show for all $y \geq \mu\sqrt{2n}$,

$$\begin{aligned}
& \Pr_{x \in \{-1,1\}^{2n}}[|x| - \alpha \cdot \sqrt{2n} \geq y] \leq \frac{1}{2} \Pr[F_{2n} \geq y] \\
\Longleftarrow & \Pr[|B_{2n}| \geq y + \alpha \cdot \sqrt{2n}] \leq \frac{1}{2} \Pr[F_{2n} \geq y] \\
\Longleftarrow & \Pr[|B_{2n}| \geq (z+\alpha) \cdot \sqrt{2n}] \leq \frac{1}{2} \Pr[F_{2n} \geq z \cdot \sqrt{2n}] \\
\Longleftarrow & 2 \cdot \exp\left(-\frac{(z+\alpha)^2}{2}\right) \leq \frac{1}{2} \Pr[F_{2n} \geq z \cdot \sqrt{2n}]
\end{aligned}$$

where the last line follows from Theorem 2.2, and in the second last line we substitute $z := y/\sqrt{2n}$.

Thus, we need to show for all $z \geq \mu$,

$$4 \cdot \exp\left(-\frac{(z+\alpha)^2}{2}\right) \quad \leq \exp\left(-\frac{z\sqrt{2n} - \mu\sqrt{2n}}{\sigma\sqrt{2n}}\right)$$

Substituting $w := z - \mu$, we need to show for all $w \geq 0$,

$$\begin{aligned}
& \exp\left(-\frac{(w+\mu+\alpha)^2}{2} + \frac{w}{\sigma}\right) && \leq 0.25 \\
\Longleftarrow & -\frac{(w+\mu+\alpha)^2}{2} + \frac{w}{\sigma} && \leq -1.4
\end{aligned}$$

The left hand side is maximized at $w + \mu + \alpha = 1/\sigma$ and for that value of w the inequality is given by

$$\frac{-1}{2\sigma^2} + \frac{1/\sigma - \mu - \alpha}{\sigma} \leq -1.4 \Longleftarrow \mu + \alpha \geq 1.4\sigma + \frac{0.5}{\sigma}$$

Also, recall that to bound the first term we needed $-\frac{\mu}{\alpha} \geq 2.47$. Let's set $\mu := -2.47\sigma$. Then we need

$$\alpha \geq (1.4 + 2.47)\sigma + \frac{0.5}{\sigma} = 3.87\sigma + \frac{0.5}{\sigma}$$

The right hand side is minimized at $\sigma = \frac{1}{\sqrt{2 \cdot 3.87}} \approx 0.36$, and substituting we get that $\alpha = 2.8$ is feasible. Recall that we also needed $\mu + \alpha \geq 1$ from the base case which is already satisfied for this choice of parameters.

■

3 ALGORITHM AND RUNNING TIME

We will now prove theorem 1.5.

Proof: [Proof of Theorem 1.5] We give a simple $O(T^2)$ space and $O(T^3)$ time algorithm.

For every subinterval (i, j) of the sequence, $i, j \in [T]$ the DP table stores $P_\alpha(S_{ij})$ where S_{ij} is the subsequence of S containing bits from position i to position j, inclusive. For $i = j$, this value is always $1 - \alpha$. For $j > i$, to compute the value of $P_\alpha(S_{ij})$, we need to take the maximum over two quantities. The first quantity is $|h(S_{ij})| - \alpha \cdot \sqrt{j - i + 1}$ which corresponds to splitting the subsequence into a single interval. This can be readily computed in constant time if we pre-compute the height of every subsequence, which can be done in $O(T^2)$ space and time. The second quantity is the maximum over all $k \in \{i, i+1, \ldots, j\}$ of $P_\alpha(S_{ik}) + P_\alpha(S_{kj})$. This corresponds to splitting the subsequence at k and then recursively computing the best payoff in each of the two intervals created. This quantity can be computed in time $j - i + 1$ since for each k we just need to read off the appropriate values ($P_\alpha(S_{ik})$ and $P_\alpha(S_{kj})$) from the DP table. ■

Next we prove Theorem 1.7

Proof: [Proof of Theorem 1.7]

Let $X \in \{-1, 1\}^T$ be the input sequence we are required to predict. Using Observation 1.6, it is easy to see that the following algorithm achieves payoff $P_\alpha(X)$ in expectation. For every $t \in \{0, 1, \ldots, T-1\}$:

1. Let $s \in \{-1, 1\}^t$ be the sequence of bits seen so far.
2. Let U_t be a sequence drawn uniformly at random from $\{-1, 1\}^{T-t-1}$ (independently for each t). Let $s_1 := s \cdot 1 \cdot U$ and $s_{-1} := s \cdot (-1) \cdot U$.
3. Make the prediction $\tilde{b} := (P_\alpha(s_1) - P_\alpha(s_{-1})/2$ for the next bit.

The key idea is that we will draw a random sequence of length T and use its suffix of length $\{-1, 1\}^{T-t-1}$ as U_t. In advance we pre-compute enough information to make the prediction as fast as possible. For each $t \in \{0, 1, \ldots, T-1\}$ we pre-compute the following information for each U_t:-

1. $h(U_t^1)$ for every prefix U_t^1 of U_t
2. $P_\alpha(U_t^2)$ for every suffix U_t^2 of U_t

The pre-computation takes $O(T^2)$ time as P_α is computed only for each suffix.

Let's describe how to use this pre-computed information to compute $P_\alpha(s_1)$ at time t (the computation of $P_\alpha(s_{-1})$ is similar). Let $1 \leq i \leq t$ and $t + 2 \leq j \leq T$. Then it is easy to check that

$$P_\alpha = \max_{i,j}(P_\alpha(s_{1i}) + P_\alpha(U_{jT}) \\ + \left|h(s_{(i+1)t})\right| + \left|h(U_{(t+1)(j-1)})\right| - \alpha \cdot \sqrt{j - i - 1})$$

Here for a sequence S, S_{ij} is the subsequence of S containing bits from position i to position j, inclusive. Note that we think of U_t as being indexed from $t + 1$ to T where the $(t+1)^{th}$ bit is 1 (since we are dealing with s_1). The second and fourth term are part of our pre-computation. The first and third terms can be computed on the fly and stored in the table as we increase t from 1 to T. Thus, for each i and j we can compute this expression in constant time and hence we can produce a prediction in $O(T^2)$ time per step.

The second part of the theorem is proved in a similar manner by using only aligned intervals for splitting the sequence (Definition 2.3) and observing that the number of aligned intervals spanning a given position is at most $O(\log T)$. The algorithm achieves payoff at least $P_{\alpha'}$ because of Lemma 1.8.

■

4 EXPERIMENTAL RESULTS

In this section we describe our experimental setup and findings.

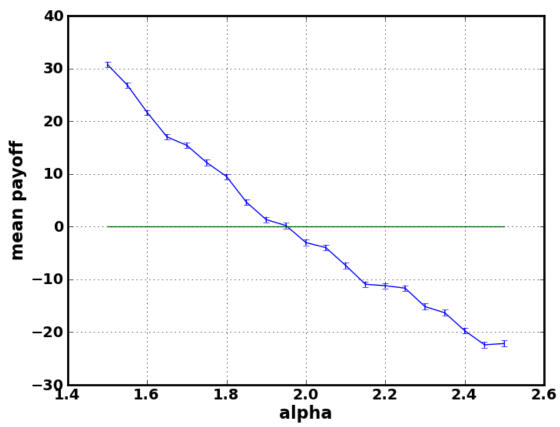
The first part of the experiment is to experimentally estimate the value of α_0. In general we may think of α_0 as a function of T. In Section 2 we saw that $\alpha_0(T)$ is bounded from above by an absolute constant for all T. In Section 4.1 below we estimate the values of α_0 for a range of T.

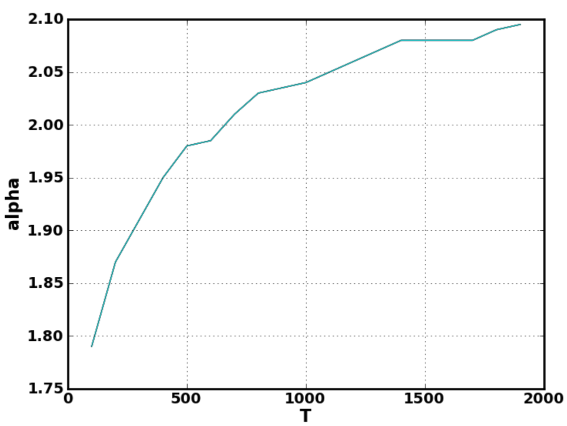
The second part of the experiment is to implement our algorithm and compare its performance against 3 other prediction algorithms. This is described in Section 4.2 below.

4.1 Computation of α_0

We denote by $\alpha_0(T)$ the minimum value of α such that the payoff function P_α is feasible for sequences of length T. For a particular T, this value can be computed using Theorem 1.5. While Theorem 1.5 requires us to compute the payoff function over all sequences of length T (to compute the expectation), we can experimentally approximate this by taking sufficiently many random sequences of length T and looking at the expectation of the sample. We are interested in $T = 389$ which is the number of minutes in a trading day for which we have returns data (there are 390 minutes in a typical trading day and the returns for the first minute is undefined).

Note that the standard error of the sample mean is obtained as the sample standard deviation divided by $\sqrt{n}$ where $n = 400$ is the number of trials. The chart on the left shows the mean payoff and standard error for various values of α for $T = 389$.





From the figure we see that $\alpha = 1.96$ is a good estimate for $\alpha_0(T)$ for $T = 389$. The figure on the right shows estimated values of α_0 for various T.

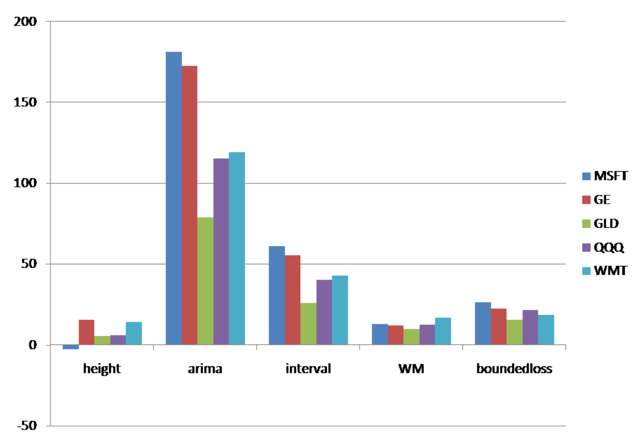
4.2 Comparison of predictive performance

The algorithms we consider are:-

1. The baseline buy and hold strategy that achieves payoff equal to the height (height)
2. The algorithm described in this paper (interval)
3. Weighted Majority algorithm (WM)
4. The algorithm of [KP11] (Algorithm 4, section 5) (boundedloss)
5. An algorithm based on Auto Regressive Integrated Moving Average (arima)

Note that algorithms 2-4 are based on ideas from regret minimization with provable guarantees while the fifth is a commonly used model for predicting time series data. To implement the fourth algorithm we use the function AUTO.ARIMA() in R which is part of the library FORECAST.

The prediction task we consider is to predict the next minute returns for a stock over a single trading day using only the previous 1 minute returns of the given stock for the given day. More precisely, we define the price of a stock at a given time taking the average of the best bid price and best ask price at that time as reported by the New York Stock Exchange (NYSE). We perform this prediction experiment over 189 days for the following 5 US stocks/ETFs from various sectors: MSFT, GE, GLD, QQQ and WMT. This gives us performance data for each algorithm for a total of $389 \times 189 \times 5 = 367,605$ data points. The results obtained are shown in the figure below.



We note that while our algorithm performs better in practice than other regret minimization based prediction algorithms with provable guarantees, it is outperformed by the ARIMA model.

References

[AB09] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. *COLT*, 2009. 1

[ACBFS02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32:48–77, 2002. 1

[AHKS06] A. Agarwal, E. Hazan, S. Kale, and R.E. Schapire. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on Machine learning*, pages 9–16. ACM, 2006. 1.1

[ALW06] J. Abernethy, J. Langford, and M. Warmuth. Continuous experts and the binning algorithm. *Learning Theory*, pages 544–558, 2006. 1.1

[AWY08] J. Abernethy, M.K. Warmuth, and J. Yellin. Optimal strategies from random walks. In *Proceedings of The 21st Annual Conference on Learning Theory*, pages 437–446. Citeseer, 2008. 1.1

[Blu97] A. Blum. Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. *Machine Learning*, 26(1):5–23, 1997. 1.1

[BM07] A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, pages 1307–1324, 2007. (document), 1, 1.1

[CBFH+97] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997. 1.1, 1.1

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games*. Cambridge University Press, 2006. 1, 1.1

[Cov65] T. Cover. Behaviour of sequential predictors of binary sequences. *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, 1965. 1, 1.1, 1.6, 1.1

[Cov91] T. Cover. Universal portfolios. *Mathematical Finance*, 1991. 1

[EDKMW08] E. Even-Dar, M. Kearns, Y. Mansour, and J. Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72:21–37, 2008. 1.1

[FSSW97] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. *STOC*, pages 334–343, 1997. (document), 1, 1.1

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 2.2

[HS09] E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. *ICML*, pages 393–400, 2009. 1, 1.1

[HSSW98] 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. 1.1

[HW98] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998. 1.1

[KP11] Michael Kapralov and Rina Panigrahy. Prediction strategies without loss. In *NIPS*, pages 828–836 (full version is available at http://arxiv.org/abs/1008.3672), 2011. 1, 1, 1.1, 4

[LW89] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *FOCS*, 1989. (document), 1

[MS08] I. Mukherjee and R. Schapire. Learning with continuous experts using drifting games. In *Algorithmic Learning Theory*, pages 240–255. Springer, 2008. 1.1

[PP13] Rina Panigrahy and Preyas Popat. Optimal amortized regret in every interval. *arXiv preprint arXiv:1304.7577*, 2013. 1, 1

[RST10] A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Beyond regret. *arXiv preprint arXiv:1011.3168*, 2010. 1.1

[Vov98] V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 1998. 1.1

[Vov99] V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, pages 247–282, 1999. (document), 1, 1.1