Sequential Investment and Online Prediction

Tengyuan Liang[1]

[1] The University of Chicago
Booth School of Business

DLA Lecture 4: No Regret

Readings: Cesa-Bianchi and Lugosi [2], Chapter 9 and 10.

[2] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006

Contents

1 *Sequential Portfolio Selection*

Consider the problem of sequential investment. A market consists of m stocks in which, in each trading period $t \in 1, 2 \ldots n$, the price of a stock may vary in an arbitrary way. An investor seeks to sequentially allocate the portfolio in a trading period of n days.

We deliberately avoid any statistical assumptions about the nature of the stock market, and evaluate the investor's wealth relative to the performance achieved by the best strategy in a class of reference investment strategies, or the so-called experts. We assume no transaction costs.

Let's fix the notations.

- **Market information**: $\mathbf{x} = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m_{\geq 0}$ denotes the ratio of closing to opening price of the i-*th* stock in that period. Since we consider a trading period $t = 1, 2, \ldots, n$, we stack the market information at time t to be the matrix $\mathbf{x}^t := [\mathbf{x}_1, \ldots, \mathbf{x}_t] \in \mathbb{R}^{m \times t}$. Again, the i-th component of $\mathbf{x}_t$ is denoted as $x_{i,t}$, is the factor by which the wealth invested in stock i increases in the period t.

- **Investment strategy**: an investment vector $\mathbf{q} = (q_1, \ldots, q_m) \in \Delta_m$ the probability simplex denotes an allocation of the total wealth to each stock. If invest a unit amount to the market vector $\mathbf{x}$ with investment strategy $\mathbf{q}$, by the end of the trading, the wealth grows to $\sum_{i=1}^{m} q_i x_i$.

A sequential investment strategy denotes a mapping from past market information to a probability distribution over the stocks, namely $\mathbf{Q}_t : \mathbb{R}_{\geq 0}^{m\times(t-1)} \to \Delta_m$, which specifies

$$\mathbf{x}^{t-1} \mapsto \mathbf{Q}_t(\mathbf{x}^{t-1}) \in \Delta_m,\ t = 1, 2, \ldots, n \tag{1.1}$$

We denote $Q_{i,t}(\mathbf{x}^{t-1})$ to denote the portation of total wealth invested to stock i in period t, based on market information collected in the past $t-1$ periods. We call the sequential investment strategy (for periods n) $\mathbf{Q} = (\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_n)$ to be the collection of such maps.

- **Wealth factor**: given the martket information $\mathbf{x}^n$ and a sequential investment stragegy $\mathbf{Q}$, compute the wealth factor

$$S_n(\mathbf{Q}, \mathbf{x}^n) := \prod_{t=1}^{n} \sum_{i=1}^{m} Q_{i,t}(\mathbf{x}^{t-1}) x_{i,t} \tag{1.2}$$

Now, we consider two sets of simple strategies.

Example (Buy-and-hold stragegy). *A buy-and-hold stragegy refers to a no-trading strategy indexed by $q \in \Delta_m$, where one distributes wealth according to q and hold it for a period of n,*

$$S^{\text{BnH}}(\mathbf{q}, \mathbf{x}^n) := \sum_{i=1}^{m} \mathbf{q}_i \prod_{t=1}^{n} x_{i,t} \tag{1.3}$$

Example (Constantly rebalancing strategy). *A constantly rebalancing strategy refers to a time-homogeneous strategy that $\mathbf{Q}_t(\mathbf{x}^{t-1}) := \mathbf{q}$ regardless of time t and the past market behavior $\mathbf{x}^{t-1}$. Note there we trade each period, thus bearing the name constantly rebalancing (as opposed to the buy-and-hold strategy, where one does not trade at all).*

$$S_n^{\text{ReB}}(\mathbf{q}, \mathbf{x}^n) := \prod_{t=1}^{n} \sum_{i=1}^{m} \mathbf{q}_i x_{i,t}\ . \tag{1.4}$$

We call the constantly rebalancing strategy

$$\mathcal{B}^{\text{ReB}} := \{(\mathbf{q}, \mathbf{q}, \ldots, \mathbf{q}), \mathbf{q} \in \Delta_m\}\ . \tag{1.5}$$

To better constrast the buy-and-hold strategy and the constantly rebalacing strategy, consider a simple case where there are two stocks,

$$\mathbf{x}^n := \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\ 0.5 & 2 & 0.5 & 2 & \ldots & 0.5 & 2 \end{bmatrix} \tag{1.6}$$

Then for n even

$$S^{\text{BnH}}(\mathbf{q}, \mathbf{x}^n) = 1, \forall q$$
$$S_n^{\text{ReB}}((1/2, 1/2), \mathbf{x}^n) = (\frac{9}{8})^{n/2}$$

1.1 Minimax Wealth Ratio

Recall the goal of the investor is to compete with a certain class of investment strategies $\mathcal{Q}$, regardless of the market behavior. Such classes might include, actively managed portfolios built based on these m stocks, or all constantly rebalancing strategies.

Definition 1 (Minimax wealth ratio)**.** *The* ***worst-case logarithmic wealth ratio*** *of a strategy* $\mathbf{P}$ *relative to a class* $\mathcal{Q}$ *is defined as*

$$W_n(\mathbf{P}, \mathcal{Q}) := \sup_{\mathbf{x}^n} \sup_{\mathbf{Q} \in \mathcal{Q}} \log \frac{S_n(\mathbf{Q}, \mathbf{x}^n)}{S_n(\mathbf{P}, \mathbf{x}^n)} \tag{1.7}$$

and define the ***minimax wealth ratio*** *as*

$$W_n(\mathcal{Q}) := \inf_{\mathbf{P}} W_n(\mathbf{P}, \mathcal{Q}) \; . \tag{1.8}$$

A few remarks follow

- For a good class of reference strategies and good market conditions, we can expect the wealth factor grow exponentially

$$\sup_{\mathbf{Q} \in \mathcal{Q}} \quad \log S_n(\mathbf{Q}, \mathbf{x}^n) = r \cdot n \tag{1.9}$$

  For instance, the S&P500 has a growth rate of $\exp(\log(1.095)n)$, where $r = 0.04$.

- If a strategy $\mathbf{P}$ satisfies

$$W_n(\mathbf{P}, \mathcal{Q}) = o(n)$$

  then it implies that $\mathbf{P}$ can compete with the class $\mathcal{Q}$

$$\log S_n(\mathbf{P}, \mathbf{x}^n) \geq \underbrace{\log S_n(\mathbf{Q}, \mathbf{x}^n)}_{=r \cdot n} - o(n) \; .$$

  The name no-regret alludes to the above property.

Example (Compete with N portfolio managers/experts)**.** *Consider* $\mathcal{Q}$ *to be the class of strategies presented by N portfolio managers,*

$$\mathcal{Q} := \{\mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(N)}\}$$

Then then simple average strategy $\mathbf{P}$ *which divide the initial wealth in* $1/N, 1/N, \ldots, 1/N$ *and then invest on each expert using the buy-and-hold strategy, we have*

$$S_n(\mathbf{P}, \mathbf{x}^n) = \frac{1}{N}\sum_{j=1}^{N} S_n(\mathbf{Q}^{(j)}, \mathbf{x}^n)$$

and thus

$$W_n(\mathbf{P}, \mathcal{Q}) \leq \log(N)$$

2 *Digression: Online Prediction with Log Loss*

Consider the online probability assignment question, where there are m items $\mathcal{Y} = \{1, 2, \ldots, m\}$ and n-periods. One wishes to maximum the likelihood of the sequence by assigning the probabilities.

- **Sequences**: A sequence of numbers is revealed sequentially $(y_1, y_2, \ldots, y_n)$, where each $y_t \in \mathcal{Y}$, finite m-items. We denote $y^t := (y_1, y_2, \ldots, y_t)$.
- **Experts**: An expert $q = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n)$ is a sequence of functions $\mathbf{q}_t : \mathcal{Y}^{t-1} \to \Delta_m$

$$y^{t-1} \mapsto \mathbf{q}_t(y^{t-1}) \in \Delta_m \tag{2.1}$$

  is a probability assignment vector, with each component

$$\mathbf{q}_t(y^{t-1}) := [q_t(1|y^{t-1}), q_t(2|y^{t-1}), \ldots, q_t(m|y^{t-1})].$$

  We denote the class of experts as $\mathcal{E}$.

- **Forecaster**: $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$ is a sequence of probability vectors similar to the definition in experts.
- **Likelihood**: One more notation, given a forecaster p (or similarly for an expert q), we define the likelihood for the sequence y^n

$$p_n(y^n) := \prod_{t=1}^{n} p_t(y_t|y^{t-1}) \ . \tag{2.2}$$

  Note here the goal is to maximize the log-likelihood in comparison to a class of Experts $\mathcal{E}$

$$M_n(\mathbf{p}, \mathcal{E}) := \max_{y^n \in \mathcal{Y}^n} \left\{ \sup_{\mathbf{q} \in \mathcal{E}} \log q_n(y^n) - \log p_n(y^n) \right\} \tag{2.3}$$

2.1 Connection to Maximum Likelihood Estimation

Definition 2 (Minimax Optimal Forecaster). *Define the minimax regret for the class $\mathcal{E}$ as*

$$M_n(\mathcal{E}) := \inf_{\mathbf{p}} M_n(\mathbf{p}, \mathcal{E})$$

The infimum turns out to be attained by the normalized maximum likelihood probability distribution (MLE)

$$p_n^\star(y^n) := \frac{\sup_{\mathbf{q}\in\mathcal{E}} q_n(y^n)}{\sum_{z^n\in\mathcal{Y}^n} \sup_{\mathbf{q}\in\mathcal{E}} q_n(z^n)}$$

and the sequential forecaster is thus defined as

$$p_t^\star(i|y^{t-1}) := \frac{p_t^\star(y^{t-1}i)}{p_{t-1}^\star(y^{t-1})} . \tag{2.4}$$

Note that this minimax optimal forecaster is computationally expansive to calculate, even for $m = 2$.

Theorem 1 (Regret for Minimax Optimal Forecaster). *The minimax regret is achieved by the minimax optimal forecaster, and*

$$M_n(\mathcal{E}) = \log\left(\sum_{y^n\in\mathcal{Y}^n} \sup_{\mathbf{q}\in\mathcal{E}} q_n(z^n)\right) \tag{2.5}$$

How to deal with the computation then? The trick is to swap the "sup" by "$\int$" over a distribution of q, denoted by $\mu \in \mathcal{P}(\mathcal{E})$.

$$\begin{aligned} p_n^\mu(y^n) &:= \frac{\int_{q\in\mathcal{E}} q_n(y^n)\,\mathrm{d}\mu(q)}{\sum_{z^n\in\mathcal{Y}^n} \int_{q\in\mathcal{E}} q_n(z^n)\,\mathrm{d}\mu(q)} \\ &= \frac{\int_{q\in\mathcal{E}} q_n(y^n)\,\mathrm{d}\mu(q)}{\int_{q\in\mathcal{E}} (\sum_{z^n\in\mathcal{Y}^n} q_n(z^n))\,\mathrm{d}\mu(q)} \quad \text{Fubini's theorem} \\ &= \int_{q\in\mathcal{E}} q_n(y^n)\,\mathrm{d}\mu(q) \end{aligned}$$

It turns out this integration swapping supermum idea also has a nice Bayesian interpretation, which we will discuss now.

2.2 Connection to Bayesian Algorithm

Definition 3 (Bayesian Mixture Forecaster). *Given a prior distribution, which is a mixture of experts in $\mathcal{E}$, denoted by μ, define the following Bayesian mixture forecaster*

$$p_n^\mu(y^n) := \int_{q\in\mathcal{E}} q_n(y^n)\,\mathrm{d}\mu(q) \tag{2.6}$$

and correspondingly

$$p_t^{\mu}(i|y^{t-1}) := \frac{\int_{q\in\mathcal{E}} q_t(y^{t-1}i)\,\mathrm{d}\mu(q)}{\int_{q\in\mathcal{E}} q_{t-1}(y^{t-1})\,\mathrm{d}\mu(q)} = \underset{posterior}{\mathbb{P}}(i|y^{t-1}) \tag{2.7}$$

where the posterior probability is calculated based on the prior $q \sim \mu$, and then $y^t \sim q_n(y^n)$.

Theorem 2 (Regret for Bayesian Mixture Forecaster)**.** *The minimax regret is achieved by the minimax optimal forecaster, and*

$$M_n(p^{\mu},\mathcal{E}) = \sup_{y^n\in\mathcal{Y}^n} \log \frac{\sup_{q\in\mathcal{E}} q_n(y^n)}{\int_{q\in\mathcal{E}} q_n(y^n)\,\mathrm{d}\mu(q)} \tag{2.8}$$

Definition 4 (Constant Experts)**.** *If the experts are time-homogeneous, namely, $q_t(i|y^{t-1}) \equiv q(i), \forall t$, we call them constant experts. For this problem, we define the constant experts class as*

$$\mathcal{E}^{\mathrm{h}} := \{(\underbrace{\mathbf{q},\mathbf{q},\ldots,\mathbf{q}}_{n}) \mid \mathbf{q}\in\Delta_m\} \tag{2.9}$$

Theorem 3 (Regret Bounds: Krichevsky-Trofimov mixture forecaster vs. minimax optimal forecaster)**.** *Consider the case of constant experts, then the minimax optimal forecaster achieves the regret*

$$M_n(\mathcal{E}^{\mathrm{h}}) = \frac{m-1}{2}\log\frac{n}{2\pi} + \log\frac{\Gamma(1/2)^m}{\Gamma(m/2)} + o_n(1)\ . \tag{2.10}$$

Pick $\mu(\mathbf{q})$ where $\mathbf{q}\in\Delta_m$ and μ being the Dirichlet prior

$$\mathrm{d}\mu(\mathbf{q}) = \frac{\Gamma(m/2)}{\Gamma^m(1/2)}\prod_{i=1}^{m}\frac{1}{\sqrt{q(i)}}\,\mathrm{d}\mathbf{q} \tag{2.11}$$

and consider the Bayesian mixture forecaster defined in (2.6)

$$p_n^{\mu}(y^n) := \int_{\Delta_m}\prod_{t=1}^{n} q(y_t)\,\mathrm{d}\mu(\mathbf{q}) \tag{2.12}$$

Then the corresponding regret bound holds

$$M_n(p^{\mu},\mathcal{E}^{\mathrm{h}}) := \sup_{y^n\in\mathcal{Y}^n}\sup_{q\in\mathcal{E}^{\mathrm{h}}}\left\{\log q_n(y^n) - \log p_n^{\mu}(y^n)\right\} \tag{2.13}$$

$$\le \frac{m-1}{2}\log\frac{n}{2\pi} + \log\frac{\Gamma(1/2)^m}{\Gamma(m/2)} + \frac{m-1}{2}\log 2 + o_n(1)\ . \tag{2.14}$$

A few remarks follow

- The Krichevsky-Trofimov mixture forecaster exceeds the minimax optimal bound just by a constant factor (independent of n). In particular,

$$M_n(p^{\mu},\mathcal{E}^{\mathrm{h}}) \le \underbrace{\inf_p M_n(p^{\mu},\mathcal{E}^{\mathrm{h}})}_{\asymp\frac{m-1}{2}\log n} + \frac{m-1}{2}\log 2 \tag{2.15}$$

- The proof is based on Gamma functions and Stirling approximations.

TL: Leave it as homework. It is not hard to show.

- The Krichevsky-Trofimov mixture may be easily calculated by a smoothed version of empirical frequencies

$$p_t^{\mu}(i|y^{t-1}) = \frac{t_i + 1/2}{t - 1 + m/2} \tag{2.16}$$

where t_i denotes the number of occurrences of i in y^{t-1} with $\sum_{i=1}^m t_i = t - 1$.

3 *Reduction: Sequential Investment to Online Prediction*

In this section, we will show a reduction to relate the two problems: sequential investment, and online probability assignment.

The idea is simple based on two steps:

- For a minimax sequential investment problem, restricted **only a strict subset of market conditions** (called Kelly market vectors), we reduce to a minimax online probability assignment problem. This will show the minimax sequential investment problem is strictly harder than the online prediction problem.
- Given any algorithm p that solves the online probability assignment problem, we induce a corresponding sequential investment algorithm $\mathbf{P}$ for **any market conditions**. This step hopes to upper bound the regret of $\mathbf{P}$ for the sequential investment problem by the regret of p for the online learning problem.

4 *Universal Portfolio and Bayesian Mixtures*

Define the initial wealth $S_0 \equiv 1$ and define the wealth at t

$$S_t^{\text{ReB}}(\mathbf{q}, \mathbf{x}^t) := \prod_{s=1}^{t} (\sum_{i=1}^{m} \mathbf{q}_i x_{i,s}) \tag{4.1}$$

Define **the universal portfolio strategy** $\mathbf{P}_{\text{UP}}^{\mu}$ induced by μ

$$P_{i,t}^{\text{UP}}(\mathbf{x}^{t-1}) := \frac{\int_{\Delta_m} \mathbf{q}_i \cdot S_{t-1}^{\text{ReB}}(\mathbf{q}, \mathbf{x}^{t-1}) \, \mathrm{d}\mu(\mathbf{q})}{\int_{\Delta_m} S_{t-1}^{\text{ReB}}(\mathbf{q}, \mathbf{x}^{t-1}) \, \mathrm{d}\mu(\mathbf{q})} \tag{4.2}$$

In any case, the universal portfolio is a weighted average of the strategies in $\mathcal{B}^{\text{ReB}}$, weighted by their past performance.

Observe that

$$
\begin{aligned}
S_n(\mathbf{P}^\mu, \mathbf{x}^n) &= \prod_{t=1}^{n} \sum_{i=1}^{m} P_{i,t}(\mathbf{x}^{t-1}) x_{i,t} \\
&= \prod_{t=1}^{n} \frac{\int_{\Delta_m} \sum_{i=1}^{m} \mathbf{q}_i x_{i,t} \cdot S_{t-1}^{\mathrm{ReB}}(\mathbf{q}, \mathbf{x}^{t-1})\, \mathrm{d}\mu(\mathbf{q})}{\int_{\Delta_m} S_{t-1}^{\mathrm{ReB}}(\mathbf{q}, \mathbf{x}^{t-1})\, \mathrm{d}\mu(\mathbf{q})} \\
&= \prod_{t=1}^{n} \frac{\int_{\Delta_m} S_{t}^{\mathrm{ReB}}(\mathbf{q}, \mathbf{x}^{t})\, \mathrm{d}\mu(\mathbf{q})}{\int_{\Delta_m} S_{t-1}^{\mathrm{ReB}}(\mathbf{q}, \mathbf{x}^{t-1})\, \mathrm{d}\mu(\mathbf{q})} \\
&= \int_{\Delta_m} S_{n}^{\mathrm{ReB}}(\mathbf{q}, \mathbf{x}^{n})\, \mathrm{d}\mu(\mathbf{q}) = \int_{\Delta_m} \prod_{t=1}^{n} (\sum_{i=1}^{m} \mathbf{q}_i x_{i,t})\, \mathrm{d}\mu(\mathbf{q}) \\
&= \int_{\Delta_m} \sum_{y_n \in \mathcal{Y}^n} \prod_{t=1}^{n} \mathbf{q}_{y_t} x_{y_t,t}\, \mathrm{d}\mu(\mathbf{q}) \\
&= \sum_{y_n \in \mathcal{Y}^n} (\prod_{t=1}^{n} x_{y_t,t}) \underbrace{\int_{\Delta_m} \prod_{t=1}^{n} \mathbf{q}_{y_t}\, \mathrm{d}\mu(\mathbf{q})}_{p_n^{\mu}(y^n)}
\end{aligned}
$$

which is governed by the online prediction problem.

```
# Algorithm: universal portfolio
# Input: a sequence of market vectors X_{i,t}, i from m assets, t from n periods
# Output: Krichevsky-Trofimov mixture forecaster to build universal portfolio
def log_wealth_factor_rebalance(Q, X):
    # m, n = np.shape(X)
    # Nsim, m = np.shape(Q)
    lnW = np.log(np.matmul(Q, X))
    lnWcum = np.cumsum(lnW, axis = 1)
    return lnWcum

def universal_portfolio(X, Nsim = 1e5):
    m, n = np.shape(X)
    alpha = 0.5 * np.ones(m)
    Q = np.random.dirichlet(alpha, int(Nsim))

    lnWcum = log_wealth_factor_rebalance(Q, X)
    P_un_normalized = np.matmul(np.transpose(Q), np.exp(lnWcum-np.log(Nsim)))
    norm_Const = np.sum(np.exp(lnWcum-np.log(Nsim)), axis = 0, keepdims=True)
    return P_un_normalized/norm_Const
```

Theorem 4 (Performance of Universal Portfolio)**.** *The universal portfolio algorithm $\mathbf{P}_{\mathrm{UP}}^{\mu}$ above satisfies the following regret guarantee over the class of constantly rebalancing strategies defined in* (1.5),

$$
\sup_{\mathbf{x}^n} \sup_{\mathbf{Q} \in \mathcal{B}^{\mathrm{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n) - \log S_n(\mathbf{P}_{UP}^{\mu}, \mathbf{x}^n) \leq \frac{m-1}{2} \log \frac{n}{2\pi} + \log \frac{\Gamma(1/2)^m}{\Gamma(m/2)} + \frac{m-1}{2} \log 2 + o_n(1)
$$

Proof. The proof hinges on a reduction to the online probability assignment problem, as hinted above.

Note that for any $\mathbf{Q} \in \mathcal{B}^{\text{ReB}}$, it naturally induced an online probability assignment problem, and therefore: Given $\mathbf{x}^n$, let $\mathbf{q}^\dagger$ be the maximizer of $\sup_{\mathbf{Q} \in \mathcal{B}^{\text{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n)$, then

$$\begin{aligned}
&\sup_{\mathbf{Q} \in \mathcal{B}^{\text{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n) - \log S_n(\mathbf{P}^\mu_{UP}, \mathbf{x}^n) \\
&= \log S_n(\mathbf{Q}^\dagger, \mathbf{x}^n) - \log S_n(\mathbf{P}^\mu_{UP}, \mathbf{x}^n) \\
&= \log \frac{\sum_{y_n \in \mathcal{Y}^n} (\prod_{t=1}^n x_{y_t,t}) \prod_{t=1}^n \mathbf{q}^\dagger_{y_t}}{\sum_{y_n \in \mathcal{Y}^n} (\prod_{t=1}^n x_{y_t,t}) \int_{\Delta_m} \prod_{t=1}^n \mathbf{q}_{y_t} \, \mathrm{d}\mu(\mathbf{q})} \\
&= \log \frac{\sum_{y_n \in \mathcal{Y}^n} (\prod_{t=1}^n x_{y_t,t}) \cdot q^\dagger_n(y^n)}{\sum_{y_n \in \mathcal{Y}^n} (\prod_{t=1}^n x_{y_t,t}) \cdot p^\mu_n(y^n)}
\end{aligned}$$

Denote $w(y^n, \mathbf{x}^n) := \prod_{t=1}^n x_{y_t,t} \in \mathbb{R}_{\geq 0}$ to be some non-negative weights, then the above

$$\begin{aligned}
&\log \frac{\sum_{y_n \in \mathcal{Y}^n} w(y^n, \mathbf{x}^n) \cdot q^\dagger_n(y^n)}{\sum_{y_n \in \mathcal{Y}^n} w(y^n, \mathbf{x}^n) \cdot p^\mu_n(y^n)} \\
&\leq \sup_{y_n : w(y^n, \mathbf{x}^n) > 0} \log \frac{q^\dagger_n(y^n)}{p^\mu_n(y^n)} \\
&\leq \sup_{y_n : w(y^n, \mathbf{x}^n) > 0} \sup_{\mathbf{q} \in \mathcal{E}^{\text{h}}} \log \frac{q^\dagger_n(y^n)}{p^\mu_n(y^n)} \\
&\leq \sup_{y_n \in \mathcal{Y}^n} \sup_{\mathbf{q} \in \mathcal{E}^{\text{h}}} \log \frac{q_n(y^n)}{p^\mu_n(y^n)}
\end{aligned}$$

Put things together, we have derived that the upper bound of the induced universal portfolio algorithm is bounded by that of the online Krichevsky-Trofimov mixture forecaster

$$\sup_{\mathbf{x}^n} \sup_{\mathbf{Q} \in \mathcal{B}^{\text{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n) - \log S_n(\mathbf{P}^\mu_{UP}, \mathbf{x}^n) \leq \sup_{y_n \in \mathcal{Y}^n} \sup_{\mathbf{q} \in \mathcal{E}^{\text{h}}} \log q_n(y^n) - \log p^\mu_n(y^n) \tag{4.3}$$

and by Theorem 3, we have the RHS is bounded by the desired quantity. □

Remark. *It turns out the equivalence is stronger; one can show that the sequential portfolio question is precisely as hard as the online probability assignment question, in the minimax sense.*

$$\begin{aligned}
W_n(\mathcal{B}^{\text{ReB}}) &= \inf_{\mathbf{P}} \sup_{\mathbf{x}^n} \left\{ \sup_{\mathbf{Q} \in \mathcal{B}^{\text{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n) - \log S_n(\mathbf{P}, \mathbf{x}^n) \right\} \\
&= \inf_{\mathbf{p}} \sup_{y_n \in \mathcal{Y}^n} \left\{ \sup_{\mathbf{q} \in \mathcal{E}^{\text{h}}} \log q_n(y^n) - \log p_n(y^n) \right\} \\
&= M_n(\mathcal{E}^{\text{h}})
\end{aligned}$$

The $\geq$ can be shown by considering only Kelly vectors [3] as market information. See Theorem 10.1 in [4].

[3] John L Kelly. A new interpretation of information rate. *the bell system technical journal*, 35(4):917–926, 1956

[4] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006

5 Exponentiated Gradient (EG) Portfolio

The **the universal portfolio strategy $\mathbf{P}^{\mu}_{\text{UP}}$** still involves multi-dimensional integrals in the space $\mathbf{q} \in \Delta_m$, and one has to calculate $O(mn)$ such m-dimensional simplex integrals. Though an online algorithm, the procedure is computationally heavy and relies on Monte Carlo simulations to calculate.

In this section, we consider a linearized version that is much faster to compute, based on exponentiated gradient (EG) principle. Unfortunately, unlike the universal portfolio $\mathbf{P}^{\mu}_{\text{UP}}$ which has $\log(n)$ regret, the EG strategy has $\sqrt{n}$ regret.

Consider **the exponentiated gradient (EG) strategy $\mathbf{P}^{\eta}_{\text{EG}}$**, an online algorithm. Define $P^{\text{EG}}_{i,1} = 1/m, \forall i \in [m]$ and update

$$P^{\text{EG}}_{i,t} = \frac{P^{\text{EG}}_{i,t-1} \exp\left(\frac{\eta x_{i,t-1}}{\langle \mathbf{P}^{\text{EG}}_{t-1}, \mathbf{x}_{t-1} \rangle}\right)}{\sum_{j=1}^{m} P^{\text{EG}}_{j,t-1} \exp\left(\frac{\eta x_{j,t-1}}{\langle \mathbf{P}^{\text{EG}}_{t-1}, \mathbf{x}_{t-1} \rangle}\right)} \tag{5.1}$$

Here, no integration is needed and the algorithm can be implemented exactly fast.

Motivation behind: Write out the total wealth for strategy $\mathbf{P} = (\mathbf{P}_t)_{t=1}^{n}$

$$-\log S_n(\mathbf{P}, \mathbf{x}^n) = \sum_{t=1}^{n} -\log \langle \mathbf{P}_t, \mathbf{x}_t \rangle \tag{5.2}$$

Define $\ell_t(\mathbf{P}) = -\log\langle \mathbf{P}, \mathbf{x}_t \rangle$ is a convex function in $\mathbf{P}$, and thus one can use the online EG algorithm,

$$P_{i,t} \propto P_{i,t-1} \exp(-\eta \nabla \ell_{t-1}(\mathbf{P}_{t-1})_i) \tag{5.3}$$

$$\propto P_{i,t-1} \exp\left(\frac{\eta x_{i,t-1}}{\langle \mathbf{P}_{t-1}, \mathbf{x}_{t-1} \rangle}\right) \tag{5.4}$$

therefore we derive

$$P_{i,t} = \frac{P_{i,t-1} \exp\left(\frac{\eta x_{i,t-1}}{\langle \mathbf{P}_{t-1}, \mathbf{x}_{t-1} \rangle}\right)}{\sum_{j=1}^{m} P_{j,t-1} \exp\left(\frac{\eta x_{j,t-1}}{\langle \mathbf{P}_{t-1}, \mathbf{x}_{t-1} \rangle}\right)} \tag{5.5}$$

```
# Algorithm: exponentiated gradient portfolio
def expGrad_portfolio(X, eta = 0.1):
    m, n = np.shape(X)
    P =  np.ones((m, n+1))/m
    for t in range(n):
```

```
        w = np.sum(P[:, t]*X[:, t])
        P_un_normalized = P[:, t]*np.exp(eta*X[:, t]/w)
        P[:, t+1] = P_un_normalized/np.sum(P_un_normalized)
    return P
```

Theorem 5 (Performance of Exponentiated Gradient)**.** *Assume that the price relatives $x_{i,t}$ all fall between two positive constants $c < C$. Then EG portfolio with the $\eta = \frac{c}{C}\sqrt{\frac{8\log m}{n}}$*

$$\sup_{\mathbf{x}^n \in [c,C]^{m\times n}} \sup_{\mathbf{Q}\in\mathcal{B}^{\text{ReB}}} \log S_n(\mathbf{Q}, \mathbf{x}^n) - \log S_n(\mathbf{P}_{EG}, \mathbf{x}^n) \tag{5.6}$$

$$\leq \frac{C}{c}\sqrt{\frac{n\log m}{2}} \tag{5.7}$$

The proof is a simple application of the optimistic mirror descent framework to analyze online problems with KL divergence as the Bregman divergence.

6 *Summary*

Note the difference in the wealth ratios

$$W_n(\mathbf{P}_{EG}, \mathcal{B}^{\text{ReB}}) \leq \frac{C}{c}\sqrt{\frac{n\log m}{2}} \tag{6.1}$$

$$W_n(\mathbf{P}_{UP}, \mathcal{B}^{\text{ReB}}) \leq \frac{m-1}{2}\log\frac{n}{2\pi} \tag{6.2}$$

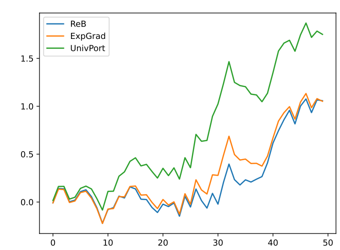

Figure 1: A simulation of the portfolios.

Let's put all the algorithms shoulder-by-shoulder as a contrast

- The MLE forecaster effectively solves the following optimization problem

$$\min_{\mathbf{q}\in\Delta_m} \sum_{t=1}^{n} -\log\langle \mathbf{q}, \mathbf{x}_t\rangle \tag{6.3}$$

- The Dirichlet mixture forecaster (universal portfolio) solves the following optimization by sampling

$$\mathbf{q} \sim \exp(-F_n(\cdot)) \tag{6.4}$$

$$F_n(\mathbf{q}) := \sum_{t=1}^{n} -\log\langle \mathbf{q}, \mathbf{x}_t\rangle - \frac{1}{2}\langle 1, \log \mathbf{q}\rangle \tag{6.5}$$

- Tthe EG forecaster solves the following optimization iteratively using online mirror descent (or linearized the problem)

$$G_n(\mathbf{q}) := \sum_{t=1}^{n} -\log\langle \mathbf{q}, \mathbf{x}_t\rangle + \frac{1}{\eta}\langle \mathbf{q}, \log \mathbf{q}\rangle \tag{6.6}$$

namely

$$\mathbf{q}_t := \arg\min_{\mathbf{q}\in\Delta_m} \langle \mathbf{q}, -\tfrac{\mathbf{x}_{t-1}}{\langle \mathbf{q}_{t-1}, \mathbf{x}_t\rangle}\rangle + \frac{1}{\eta}\text{KL}(\mathbf{q}\|\mathbf{q}_{t-1}) \tag{6.7}$$

References

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

John L Kelly. A new interpretation of information rate. *the bell system technical journal*, 35(4):917–926, 1956.