Lecture 7 — August 26

Lecturer: Aditya Gopalan *Scribe: Madhavaraj A*

7.1 Introduction

In this lecture, we will discuss about sequential portfolio allocation strategies for investing in the stock markets. The ultimate goal of the portfolio allocation algorithm is to distribute one's wealth at each trading instants among a certain number of stocks so as to get maximal return among possible classes of investment strategies.

We shall discuss about the following algorithms,
(i) Buy and hold strategy
(ii) Constantly rebalancing portfolio strategy (*CRP*)
(iii) Cover's universal portfolio algorithm

7.2 Definitions and sequential investment setting

Suppose we have n stocks in the stock market, then at every round t, we can define a market vector $X_t = (x_{1,t}, x_{2,t}, ..., x_{n,t})$ for all $t = 1, 2, 3, ...$ where,

$$x_{i,t} = \frac{\text{Closing price of stock i at the end of round t}}{\text{Opening price of stock i at the start of round t}} \geq 0$$

(i.e., Investment of Rs. 1 at round t on stock i gives a return of $x_{i,t}$ at the end of round t)
Here, we consider arbitrary $\{X_t\}$ and no assumption is made on its choice.

Alternatively, statistical models of the markets can be viewed as a Geometric random walk or Brownian motion, where, $x_{i,t} = \exp(\text{Random Walk(t)})$.

Investment decision at round t is given by, $Q_t = (Q_{1,t}, Q_{2,t}, ..., Q_{n,t}) \in \Delta_m$ such that, $\sum_{i=1}^{n} Q_{i,t} = 1$

Q_t is the fraction of current wealth invested at round t. In general,
$Q_t = Q_t(X_1, X_2, ..., X_{t-1}) = Q_t\left(X^{t-1}\right)$

Investment Algorithm Q is defined as the sequence $Q_1, Q_2, ..., Q_t$
$Q : \mathbb{R}^{m\times(t-1)} \to \Delta_m$

7.3 Buy and Hold Strategy

This is a simple algorithm where we divide the initial money according to $Q_1 \in \Delta_m$ and sit idle.

$$\begin{aligned}\text{Wealth after T rounds} &= \mathrm{S}_T(\mathrm{QX}^T) \\ &= \sum_{i=1}^{n} Q_{i,t} \prod_{t=1}^{T} x_{i,t}\end{aligned}$$

7.4 Constantly Rebalancing Portfolio Strategy

In this algorithm, the wealth at each round is equally distributed across all stocks irrespective of X^{t-1}. i.e., $Q_t\left(X^{t-1}\right) = b = (b_1, b_2, ..., b_n) \in \Delta_m$

e.g., $m = 2$, $\{X_t\} = \left(1, \frac{1}{2}\right)\ (1,2)\ \left(1, \frac{1}{2}\right)\ (1,2).....$ Let b $= \left(\frac{1}{2}, \frac{1}{2}\right)$

Then wealth after T rounds is given by,

$$\begin{aligned}S_T\left(bX^T\right) &= 1 \times \left(\tfrac{1}{2} \times 1 + \tfrac{1}{2} \times \tfrac{1}{2}\right) \times \left(\tfrac{1}{2} \times 1 + \tfrac{1}{2} \times 2\right) \times .... \\ &= \left(\tfrac{9}{8}\right)^{\frac{T}{2}}\end{aligned}$$

In the buy and hold strategy, the investor will not make any money in the above example, whereas in the *CRP* strategy, the investor makes some profit. To compare the performance of an algorithm with respect to all possible classes of strategies $\mathbb{Q}$, we define wealth ratio of an algorithm, with respect to $\mathbb{Q}$ as,

$$\begin{aligned}&= \max_{X^T} \log\left(\sup_{q \in Q} \frac{S_T(q, X^T)}{S_T(\text{Algo}, X^T)}\right) \\ &= \sup_{X^T} \sup_{q \in Q} \sum_{t=1}^{T} \log \frac{1}{\langle X_t, p_t\rangle} - \sum_{t=1}^{T} \log \frac{1}{\langle X_t, q_t\rangle}\end{aligned}$$

The above expression can also be viewed as Regret of the algorithm with following setting,
(i) Decision space $D = \Delta_m =$ all distribution on unit simplex
(ii) Observed output $y = \mathbb{R}_+^m$ $\mathrm{X}_t \in \mathbb{R}_+^m$
(iii)Loss function $l(d, y) = -\log\langle d, y\rangle$

7.5 Cover's Universal Portfolio Algorithm

Goal: To compete with all the CRPs, i.e., $\mathbb{Q}$ = CRP

Intuition: Use exponential weights algorithm (η = 1) with log-loss (as it is 1-exp-concave) over all *CRPs* in $\mathbb{Q}$.

Weight of CRP ($b \in \Delta_m$) at time t is given by,

$$
\begin{aligned}
W_t(b) &= \exp\left(-\eta \sum_{s=1}^{t-1} \log \tfrac{1}{\langle b, X_s\rangle}\right) \\
&= \prod_{s=1}^{t-1} \langle b, X_s\rangle \\
&= S_{t-1}(b, X^{t-1})
\end{aligned}
$$

Universal portfolio decision at time t is given by,

$$
\begin{aligned}
p_t &= \int_{\Delta_m} b \frac{W_t(b)}{\int_{\Delta_m} \mu\, W_t(b')\, \mathrm{db'}} \mu \, \mathrm{db} \qquad (\mu \text{ is uniform distribution on the simplex } \Delta_m) \\
&= \frac{\int_{\Delta_m} b\, \mathrm{S}_{t-1}(b, X^{t-1})\, \mu \, \mathrm{db}}{\int_{\Delta_m} \mathrm{S}_{t-1}(b', X^{t-1})\, \mu\, \mathrm{db'}}
\end{aligned}
$$

The total wealth after T rounds is expressed as,

$$
\begin{aligned}
S_T(\text{UP},\, X^T) &= \prod_{t=1}^{T} \langle p_t, X_t\rangle \\
&= \prod_{t=1}^{T} \frac{\int_{\Delta_m} \langle b, X_t\rangle\, \mathrm{S}_{t-1}(b, X^{t-1})\, \mu\, \mathrm{db}}{\int_{\Delta_m} \mathrm{S}_{t-1}(b, X^{t-1})\, \mu\, \mathrm{db}}
\end{aligned}
$$

The above telescopic series reduces to,

$$
= \int_{\Delta_m} \mathrm{S}_T(b, X^T)\, \mu\, \mathrm{db}
$$

Interpretation: Universal portfolio algorithm performs *buy and hold* strategy across all *CRP*.

Notes:

(i) UP is not efficient and is computationally infeasible as integration over Δ_m takes much time (of order $exp(m)$). However, we can make grids on Δ_m with $\left(\frac{1}{\delta}\right)^m$ grid points for computation. Efficient implementation of this algorithm is done in [1].

(ii) We will see in online convex optimization (in the next few lectures) that there exists simpler and efficient universal portfolio algorithms only with slightly worse regret.

Theorem: Regret of Cover's UP is given by,

$$
\sup_{X^T \in \mathbb{R}^{m \times T}} R_T(UP, X^T) \leqslant (m-1)\log T + \text{constant}
$$

Proof: Technique from [2]

Let the best performing *CRP* with respect to X^T be $b^* \in \Delta_m$. Then, the total wealth is given by,

$$
S_T(b^*,\, X^T) = \sup_{b \in \Delta_m} S_T(b,\, X^T)
$$

Let us define neighbourhood of b^* for any $\varepsilon > 0$ as,

$$\text{Ball}_\varepsilon(b^*) = \{(1-\varepsilon)b^* + \varepsilon b : b \in \Delta_m\}$$

Properties of Neighbourhood of ε

Lemma 1: $Volume\,(Ball_\varepsilon(b^*)) = \varepsilon^{m-1} Volume(\Delta_m)$
For any $S \subset \Delta_m,\ Volume(S) = \int\limits_S \mu\ db$

Lemma 2: $\forall b \in \text{Ball}_\varepsilon(b^*)$
$S_T(b,\ X^T) \geqslant S_T(b^*,\ X^T)(1-\varepsilon)^T$

Consider,

$$\begin{aligned} S_T(UP,\ X^T) &= \int\limits_{\Delta_m} \mathrm{S}_T(b, X^T)\ \mu\ \mathrm{db} \\ &\geqslant \int\limits_{\text{Ball}_\varepsilon(b^*)} \mathrm{S}_T(b, X^T)\ \mu\ \mathrm{db} \\ &\geqslant \int\limits_{\text{Ball}_\varepsilon(b^*)} \mathrm{S}_T(b^*, X^T)\ (1-\varepsilon)^T\ \mu\ \mathrm{db} \\ &= \mathrm{S}_T(b^*, X^T)\ (1-\varepsilon)^T\ \varepsilon^{m-1} \end{aligned}$$

Therefore,

$$\begin{aligned} \log \frac{\mathrm{S}_T(b^*, X^T)}{\mathrm{S}_T(UP, X^T)} &= \text{Regret}(UP, X^T) \\ &\leqslant (m-1)\log\frac{1}{\varepsilon} + T\log\frac{1}{1-\varepsilon} \end{aligned}$$

Setting $\varepsilon = \frac{1}{T}$ we get,

$$\text{Regret}(UP, X^T) \leqslant (m-1)\log T + T\log\left(\frac{T}{T-1}\right)$$

7.6 Next Lecture : Online convex optimization

In the next lecture, we will look at online convex optimization algorithms.

7.6.1 General Framework

(i) Rounds $t = 1, 2, 3, \ldots.$
(ii) Player plays from decision space $X \in \mathbb{R}^d$ (a convex set)
(iii) At each round t,
  (a) Play $X_t \in X$

(b) Receive a convex function, $f_t : X \to \mathbb{R}$
(c) Suffers loss, $f_t(X_t)$
(iv) The goal is to minimize regret given by,

$$R_T = \sum_{t=1}^{T} f_t(X_t) - \inf_{X^* \in X} \sum_{t=1}^{T} f_t(X^*)$$

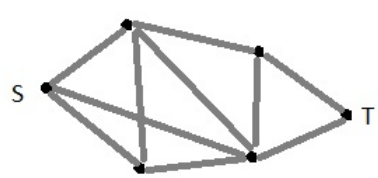
General properties of online convex optimization techniques

(a) Typically ∞ decisions.
(b) More geometric intuition
(c) Can leverage convex optimization algorithms
(d) Originated in recent times from [3]

7.6.2 Application 1: Prediction with N experts on convex decision sets

Decision space, $X = \Delta_m$
Loss function, $f_t(Z) = l\,(\langle Z, \text{ expert advice at t}\rangle\,,\ \text{y}_t)$

7.6.3 Application 2: Online shortest path



This is a combinatorial setting (which can also be viewed as N expert problem) where,
(a) Each expert is a path from S to T.
(b) In general, number of paths is exponential to the network size.
(c) Costs on the edges are chosen by nature at each step.
(d) Upon playing a path, the algorithm gets to see costs of all edges.

A naive application of Exp-Wts is expensive since there are a large number of experts. One can be much smarter and consider playing points from the space of all distributions on S-T paths, which lies in the so-called "$flow\ polytope$" (this is via network flow theory) which lies in Euclidean space of dimension $\#edges \ll \#paths$. This puts the problem in the online Convex Optimization setting.

References

[1] Adam Kalai and Santosh Vempala. 2003. Efficient algorithms for universal portfolios. J. Mach. Learn. Res. 3 (March 2003), 423-440.

[2] Avrim Blum and Adam Kalai. 1997. Universal portfolios with and without transaction costs. In Proceedings of the tenth annual conference on Computational learning theory (COLT '97). ACM, New York, NY, USA, 309-313.

[3] Martin Zinkevich. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. ICML 2003.