AI-3001: Advanced Topics in Machine Learning **Aug-Nov 2022**

Lecture 7: Portfolio Optimization

Lecturer: Ganesh Ghalme *Scribes: Ganesh Ghalme*

7.1 Today

- Buy and hold Strategy
- Constant rebalancing strategy (CRP)
- Cover's universal portfolio algorithm (CUP)

Portfolio Optimization : 1) Actions setting 2) full information 3) Uses exponential weights algorithm.

7.2 The Setting

- There are N stocks
- Each time algorithm invests the current wealth in n stocks
- Let $x_t \in \Delta_N$ be the portfolio at time t and a vector r_t denote the returns on day t. That is

$$r_{i,t} = \frac{\text{Closing price of stock } i \text{ on day } t}{\text{Opening price of stock } i \text{ on day } t} \tag{7.1}$$

- Let $r^t = (r_1, r_2, \cdots, r_t)$ be the sequence of vectors representing the market fluctuations.
- Wealth is carried forward; no transaction fees, no short trading, no buyouts
- WLOG: let the initial wealth be 1 rupee.
- A very mild assumption: No stock value shoots to infinity at any time and all stock values do not have zero returns at the same time.

7.3 Buy-and-hold strategy

Distribute the initial wealth in N stocks according to some distribution and do not make any changes after that (i.e. hold).

Observation 1. *The wealth of buy-and-hold strategy after time T satisfies*

$$W_T(x, r^T) \le \max_{i=1,2\cdots,N} \prod_{t=1}^{T} r_{i,t} \tag{7.2}$$

Proof.

$$\sum_{i=1}^{n} x_i(\prod_{t=1}^{T} r_{i,t}) \leq \max_{i=1,2,\cdots N}(\prod_{t=1}^{T} r_{i,t})$$

□

Example 7.1. *Let there are two stocks whose returns are as follows*

Stock 1	1	1	1	1	1	1	$\cdots$
Stock 2	1/2	2	1/2	2	1/2	2 $\cdots$	

Buy-and-hold strategy guarantees atmost 1 returns no matter how the initial wealth is distributed.

7.4 Constant Rebalancing portfolio (CRP)

Even in this case, we use a fixed distribution to distribute the wealth among N stocks. However, the wealth is redistributed at each time period as compared to the buy-and-hold strategy where wealth was distributed only once. Say the wealth is distributed equally among both stocks using a CRP strategy. The return after T trading rounds is given by

$$\begin{aligned} W_t(b, r^t) &= \prod_{t=1}^{T}(\sum_{i=1}^{n} b_i r_{i,t}) \\ &= \frac{3}{2} \cdot \frac{3}{4} \cdot \frac{3}{2} \cdot \frac{3}{4} \cdot \frac{3}{2} \cdot \frac{3}{4} \cdots = \left(\frac{9}{8}\right)^{T/2} >> 1 \end{aligned}$$

Consider another CRP strategy $b' = (3/4, 1/4)$. The wealth after T rounds in this case is $W_t(b, r^t) = \frac{7}{8} \cdot \frac{5}{4} \cdot \frac{7}{8} \cdot \frac{5}{4} \cdots = \left(\frac{35}{32}\right)^{T/2} < \left(\frac{9}{8}\right)^{T/2}$

7.5 Regret

Let $\mathcal{Q}$ be the set of all CRPs. Note that $\mathcal{Q} = \Delta_N$. Furthermore, let

$$b^* = \arg\max_{b \in \Delta_N} W_T(b, r^T) \tag{7.3}$$

be the best CRP. Our goal is to track the best CRP as closely as we can. The performance of the algorithm is evaluated by the regret defined similar to how we defined it so far. As we are considering the gains setting instead of loss setting here the regret of an algorithm is defined as follows,

$$\mathcal{R}_T(ALG) = \sum_{t=1}^{T} g(b^*, r_t) - \sum_{t=1}^{T} g(p_t, r_t) \tag{7.4}$$

Here the gain function is defined as

$$g(b, r_t) = \log(\langle b, r_t \rangle) \tag{7.5}$$

Equivalently we can define loss as $\ell(b, r_t) = -\log(\langle b, r_t \rangle)$

7.6 Covers Universal portfolio algorithm

Recall $W_t(b, r^t) = \prod_{t'=1}^{t} \left(\sum_{i=1}^{n} b_i r_{i,t'} \right) = \prod_{t'=1}^{t} \langle b, r_{t'} \rangle$ = total wealth after (and including) time t. Discretized version:

Algorithm 1: Discretized CUP

Input: # stocks N
Initialize: initial wealth=1, # of CRP's = m ;
for $t = 1, 2, \cdots$ **do**
  - **Play** $p_{j,t} = \frac{\sum_{k=1}^{m} b_{j,k} \cdot W_{t-1}(b_k, r^{t-1})}{\sum_{k=1}^{m} W_{t-1}(b_k, r^{t-1})}$;
  - **Observe** r_t ;
  - **Update** $W_t(b_k, r^t) = \langle p_t, r_t \rangle W_{t-1}(b_k, r^{t-1})$;

end

Continuous version plays $p_{j,t} = \frac{\int_{\Delta_N} b_j \cdot W_{t-1}(b, r^{t-1}) \cdot \mu(b) db}{\int_{\Delta_N} W_{t-1}(b, r^{t-1}) \cdot \mu(b) db}$. We will represent $p_t \in \Delta_N$ as an action of CUP. Here μ is a density function of buy-and-hold distribution over different CRP's. We will use uniform distribution of initial wealth i.e. $\mu(b) = 1$ for all $b \in \Delta_N$.

It is easy to see that p_t is a valid distribution (Fubini's Theorem).

Remarks:

1. CUP does a buy-and-hold on all CRP strategies.

2. CUP is exponential weights in disguise.

$$W_{t-1}(b, r^{t-1}) = \prod_{s=1}^{t-1} \Big(\sum_{i=1}^{n} b_i r_{i,t} \Big) = \prod_{s=1}^{t-1} \Big(\langle b, r_t \rangle \Big) = e^{-\sum_{s=1}^{t-1} \log(1/\langle b, r_t \rangle)}$$

3. Surprising fact: CUP does not assume anything about how the market evolves (except for boundedness assumption which is very weak). That is, the future stock prices may not have any correlation with the past and vary in adversarial manner. The surprising fact is even in such a model the current investment strategy depends on the past performance.

4. Universal portfolio gives at-least as much wealth as geometric value line index defined as

$$W_T(\text{CUP}, r^T) \geq \Big(\prod_{i=1}^{N} W_T(\mathbf{e}_i, r^T) \Big)^{1/N}$$

   Here, $\mathbf{e}_i$ is the N dimensional vector with ith component set as 1 and others 0.

5. Geometric Intuition: The algorithm is essentially tracking the optimal CRP. The first intuition is that the CRPs that are in the neighborhood of the optimal CRP give large returns. We hope that these returns are large enough so that they make up for the bad investment made by all other CRPs.

6. The guarantee of Uniform portfolio algorithm is tight (upto constant). That is one cannot guarantee the worst case regret that is smaller that the regret of CUP in order terms (i.e. sub-logarithmic).

Let that distribution be uniform i.e. $\mu(b) = 1$ for all $b \in \Delta_m$ (Another popular density is Dirichlet). We have,

$$p_{j,t} = \frac{\int_{\Delta_N} b_j \cdot e^{-\sum_{s=1}^{t-1} \log(1/\langle b, r_t \rangle)} db}{\int_{\Delta_N} e^{-\sum_{s=1}^{t-1} \log(1/\langle b, r_t \rangle)} db} \tag{7.6}$$

Let $\ell(x, r_t) := -\log(\langle x, r_t \rangle)$ denote the loss function. This is called a log-loss.

Claim 7.2. *Log-loss is 1-exp concave.*

The proof follows from the definition. Also, note that the log-loss is not strongly convex (prove this!).

Reading Assignment Read the article on Cover's algorithm appeared in Stanford News Service.

Lemma 7.3.

$$W_T(\text{CUP}, r^T) = \frac{\int_{\Delta_N} S_T(b, r^T) db}{\texttt{Vol}(\Delta_N)} \tag{7.7}$$

Proof. We have,

$$\begin{aligned} W_T(\text{CUP}, r^T) = \prod_{t=1}^{T} \langle p_t, r_t \rangle &= \prod_{t=1}^{T} \frac{\int_{\Delta_N} \langle b, r_t \rangle W_{t-1}(b, r^{t-1}) db}{\int_{\Delta_N} W_{t-1}(b, r^{t-1}) db} \\ &= \prod_{t=1}^{T} \Big(\frac{\int_{\Delta_N} W_t(b, r^t) db}{\int_{\Delta_N} W_{t-1}(b, r^{t-1}) db} \Big) \\ &= \frac{\int_{\Delta_N} W_T(b, r^T) db}{\texttt{Vol}(\Delta_N)} \end{aligned}$$

$\square$

Definition 7.4 (Minkowski Set)**.**

$$B_\delta(b) = (1-\delta) \cdot b + \delta \cdot \Delta_N = \{(1-\delta) \cdot b + \delta b' : b' \in \Delta_N\} \tag{7.8}$$

Claim 7.5.

$$\texttt{Vol}(B_\delta(b^*)) = \delta^{N-1} \texttt{Vol}(\Delta_N). \tag{7.9}$$

Proof follows from the fact that all the coordinates are shrunk by δ. And the object Δ_N is embedded in $N-1$ dimensional space.

Claim 7.6.

$$W_t(b, r^t) \geq (1-\delta)^T W_t(b^*, r^t) \textit{ for all } b \in B_\delta(b^*) \textit{ and for all } t. \tag{7.10}$$

Proof. Let $b \in B_\delta(b^*)$ is given. We have from the definition of $B_\delta(b^*)$ that

$$\begin{aligned} \langle b, r_t \rangle &= (1-\delta)\langle b^*, r_t \rangle + \delta \langle b', r_t \rangle \text{ for some } b^{'} \in \Delta_N \\ &\geq (1-\delta)\langle b^*, r_t \rangle \qquad \text{(The second term is non-negative)} \\ \implies W_t(b, r^t) &= \prod_{t=1}^{T} \langle b, r_t \rangle \geq (1-\delta)^T \prod_{t=1}^{T} \langle b^*, r_t \rangle = (1-\delta)^T W_t(b^*, r^t) \end{aligned}$$

There is another (more general) way to interpret this result. Let $w(b, r_t) := \langle b, r_t \rangle = e^{-\ell(b,r_t)}$. Since log loss is 1-exp concave we have that $w(b, r_t)$ is concave i.e. $w(b, r_t) \geq (1-\delta)w(b^*, r_t) + \delta w(b', r_t)$ for some $b' \in \Delta_N$ such that $b = (1-\delta)b^* + \delta b'$.[1] We have

$$W_t(b, r^t) = \prod_{t=1}^{T} w(b, r_t) \geq \prod_{t=1}^{T} \Big((1-\delta)w(b^*, r_t) + \delta w(b', r_t)\Big) \geq (1-\delta)^T \prod_{t=1}^{T} w(b^*, r_t)$$

□

The above result shows that the CRPs in the neighbourhood of best CRP do not do arbitrarily badly. That is, even if they are in a small neighbourhood they create enough wealth that the loss incurred in investment made in looser CRPs is comparatively insignificant. The second interpretation comes handy to extend the above result for any non-negative exp-concave loss functions.

With these tools we now prove the most important result of this class.

Theorem 7.7. $\mathcal{R}_T(\text{CUP}) \leq (N-1)\log(T) + 2$

Proof. Putting everything together.

$$\begin{aligned} W_T(\text{CUP}, r^T) &= \frac{\int_{\Delta_N} W_T(b, r^T) db}{\int_{\Delta_N} \texttt{Vol}(\Delta_N)} && \text{(From Eq. 7.7)} \\ &\geq \frac{(1-\delta)^T \int_{B_\delta(b^*)} W_T(b^*, r^T) db}{\int_{\Delta_N} \texttt{Vol}(\Delta_N)} && \text{(From Eq. 7.9)} \\ &= (1-\delta)^T \delta^{N-1} W_T(b^*, r^T) \end{aligned}$$

We have

$$\log\left(\frac{W_T(b^*, r^T)}{W_T(\text{CUP}, r^T)}\right) \leq (N-1)\log\left(\frac{1}{\delta}\right) + T\log\left(\frac{1}{1-\delta}\right)$$

Let $\delta = \frac{1}{T}$, we have

$$\mathcal{R}_T(\text{CUP}) = \log\left(\frac{W_T(b^*, r^T)}{W_T(\text{CUP}, r^T)}\right) \leq (N-1)\log(T) + T\log\left(1 + \frac{1}{T-1}\right)$$

The last term is small for large value of T. Hence the regret is logarithmic in T. □

7.7 Historical Note and references

The theory of Universal portfolio was first developed by Cover [Cov96]. It was a surprisingly simple and powerful algorithm that provided theoretical guarantees as well as delivered practical performance. There are a few issues though. The first is computational. [KV03] initiated study computationally efficient algorithms to implement Covers result. The second issue is that of transaction fees. Covers theoretical result assumed that there is no transaction fee and hence algorithm could rebalance the portfolio in each time period. [BK97] study the variant with transaction costs. Also see the latest papers which cite these papers.

It was later shown by [HKKA06] that the universal portfolio theory can be applied to any α-exp concave loss functions (See also Section 4.3 of the OCO book [Haz19]).

[1] Exercise: Convince yourself that such a b' must exist in Δ_N.

References

[BK97] Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, COLT '97, page 309–313, New York, NY, USA, 1997. Association for Computing Machinery.

[Cov96] Thomas M. Cover. Universal portfolios. 1996.

[Haz19] Elad Hazan. Introduction to online convex optimization, 2019.

[HKKA06] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, COLT'06, page 499–513, Berlin, Heidelberg, 2006. Springer-Verlag.

[KV03] Adam Kalai and Santosh Vempala. Efficient algorithms for universal portfolios. *J. Mach. Learn. Res.*, 3(null):423–440, mar 2003.