Assignment 2 (2024)

Day Trading in the Stock Market

Christos Kostadimas

2020030050

May 2024 , Chania

1 A few words about the Multiplicative Weights Algorithm (MWA)

The Multiplicative Weights Algorithm (MWA) is a powerful and versatile algorithm used to solve a variety of optimization and learning problems, often framed as the "experts problem." In this specific problem , the experts are the stocks.

Problem Setup:

- The MWA addresses scenarios where a decision-maker (the learner) must repeatedly choose among several options (experts) based on their past performance.
- Each expert provides a prediction or action, and the learner selects a weighted combination of these predictions or actions.

Algorithm Mechanics:

- The learner maintains a weight for each expert, representing the learner's confidence in that expert.
- Initially, all experts are given equal weight (equal to 1).
- After each round, the learner updates the weights based on the performance (loss - denoted as li in my script) of each expert. Note that i am scaling the loss, using the ***min_max_scaling*** function to take values from -1 to 1 $l_i \in [-1, 1]$.
- Weights are updated multiplicatively, meaning weights are adjusted by a factor that depends on the expert's loss and a learning rate eta.

  Let w_i^t denote the weight of expert i at round t

  Let $w_i^{(t+1)}$ denote the weight of expert i at round (t+1)

  Let l_i^t denote the loss of the expert i at round t

  The weight is updated as follows:

$$w_i^{(t+1)} = (1 - eta)^{l_i^t} \cdot w_i^t$$

Decision Making :

- The learner's decision is based on the weighted combination of the experts' recommendations.

Learning rate eta :

- The learning rate is a positive parameter that controls the extent to which the weights are adjusted in response to losses.It essentially determines the sensitivity of the algorithm to the performance of the experts.

- The optimal choice of eta depends on the specific problem and can be critical for the algorithm's performance.

- The optimal value of eta is: $\sqrt{ln(K)/T}$, where K is the number of experts (10 in our case) and T is the horizon (2000, equal to the number of days in our case). So $eta_{optimal} = 0.03$.

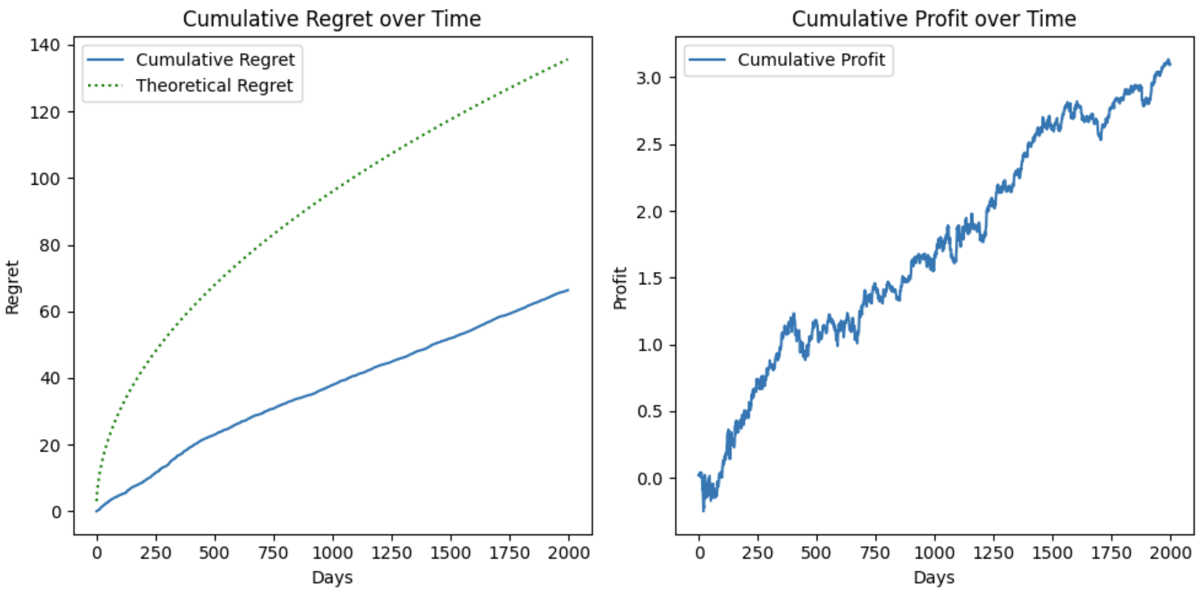
Some final important notes :

- **I chose eta=0.5 even though $eta_{optimal}$ = 0.03.** A small eta results in "slow" weight updates, making the algorithm less responsive to recent losses. The plots that came up for eta=0.03 where good, but not as good as the ones for eta=0.5 , so I simply chose eta to be 0.5.

- I did the following assumption: each day I invest 1euro E.g. if for stock 5 you see at day 10 a value +0.023 it means that if you invest 1 euro in this stock, I will get back 0.023 euros for that day.

2 Plots from the simulation

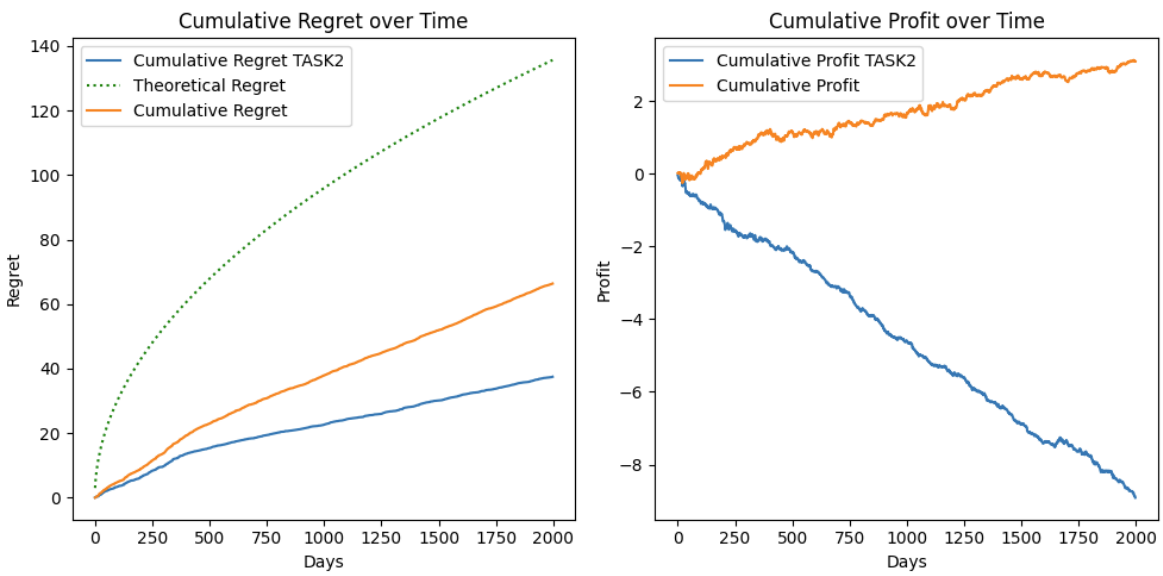
2.1 Task 1 : Experts Setup



Notes about the above figure:

- In the above figure we can see in the left side the Cumulative regret of the algorithm, from day 1 to the last day and in the right side we can see the Cumulative profits of the algorithm (i.e., how much you've made in total by day 2, day 3, etc.).
- **CUMULATIVE PROFIT:** In Task 1 the cumulative profit reaches value equal to about 3.25€ at day 2000. This means that if I invested 2000€ for 2000 days (1€ per day) , I would end with a total of 2000+3.35 = 2003.25€ . So I would earn 3.25€.
- The Cumulative Regret in this case seems to be sublinear to the theoretical one and that is a good thing to see.

2.2 Task 2: Experts with Transactions Fees



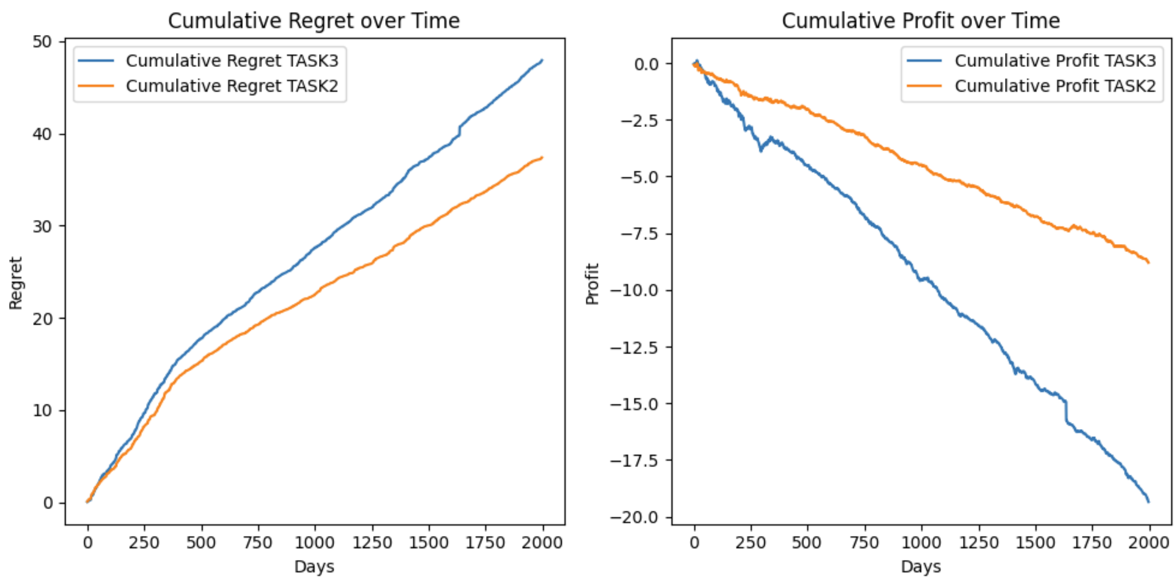
Notes about the above figure:

- We can see that the Cumulative profit is decreasing each day, and thats too pessimistic. Lets break it down...
- In Task 2 the results are more pessimistic. This was expected due to the transaction fees.

- More specifically in our case with the specific fees 0.5%, 1%, 1.5%,... for stocks 0 to K, respectively, it seems that the profits of the stocks are much smaller than the fees. (Ugh...fees are ruining our plans...)

- Lets be more specific:
  - Denote as p_i the profit that a specific stock i gives.
  - Denote as c_i the fee of that specific stock i.
  - Since I invest one euro at stock i, i will get back (p_i - c_i) €
  - If c_i is greater than p_i, I will loose money. Thats seems to be the case here..
- The fees are too "heavy" and unbearable and that makes me loose money.

2.3 Task 3: Bandits with Transaction Fees



Notes about the above figure:

- It's "wise" to expect that the performance of the Multiplicative Weights Algorithm (MWA) for experts would generally be better than the performance of MWA for bandits, though both algorithms can perform well in their respective problem domains. Here's why this difference in performance might occur.
  - In the MWA in the expert setup , the learner receives a full feedback from ALL experts at each round, allowing the update of the weights based on the performance of each expert. In fewer words the learner has access to a richer set of information , enabling to make better decisions overtime.
  - In the MWA in the badnit setup the learner receives less information. The learner receives feedback only for the chosen expert/action without information about the rest experts.The limited feedback makes it challenging to strike an optimal balance between exploration and exploitation.