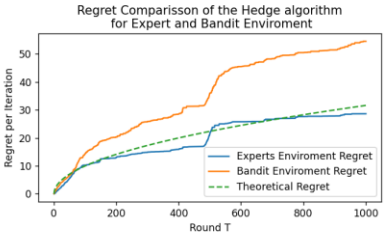
Second Exercise – Reinforcement Learning and Dynamic Optimazation

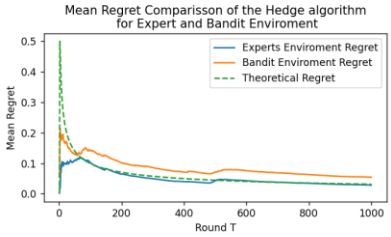
Student : Toganidis Nikos
Student id : 2018030085
Date : 25/04/2023

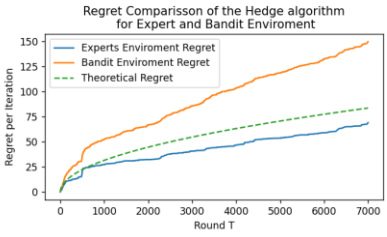
Introduction :

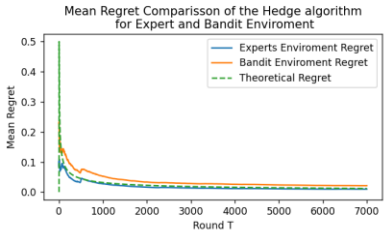
The goal of this exercise was to develop algorithms that could predict the least loaded server at every time round, given a dataset of real traffic loads over time for a number of servers. The dataset contained non-stationary demands and was normalized. More specifically, the Multiplicative Weights (*MW*) algorithm was implemented assuming an "Experts" environment where the load of other servers could be learned at every round, as well as in an (*adversarial*) "Bandits" environment. The cummulative regret of both cases was compared for horizon values T = 1000 and T = 7000 (*the entire duration of the dataset*). Additionally (*for the second part*), the Upper Confidence Bound (*UCB*) algorithm was adapted to the problem with losses instead of rewards and compared to the MW algorithm for the bandit setting (*again for T = 1000, T = 7000*).

Part 1 : For the first part of the assignment, the regrets of the Multiplicative Weights algorithm were compared for the "Experts" and "Bandit" environments, for horizon values of T = 1000 and T = 7000. The cumulative regret and mean regret of each environment was plotted against the number of rounds, along with the theoretical regret of the algorithm. The results were presented in a single graph for each horizon value, allowing for easy comparison of the algorithm's performance under different settings.





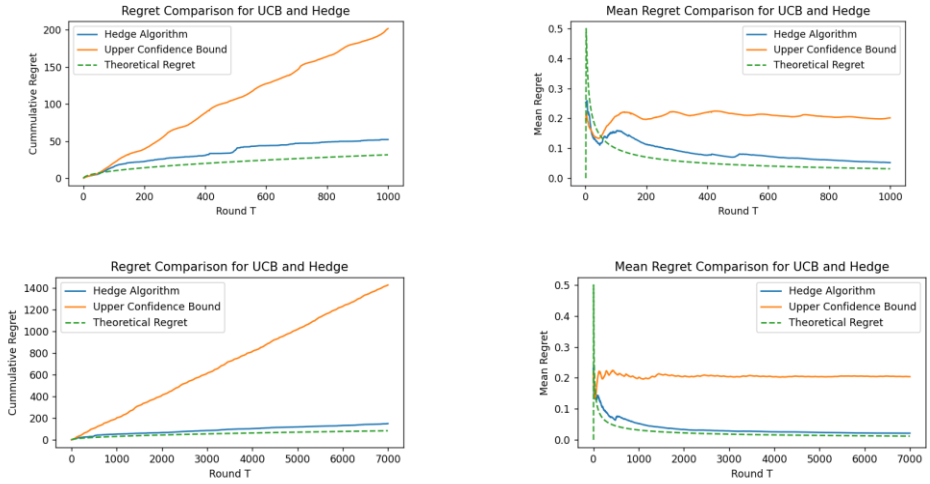




According to the graphs, it is obvious that the cumulative regret of the Hedge algorithm applied to the bandit environment is worse than the one applied to the experts' environment as expected. In the experts' environment, the algorithm can learn from the mistakes of other experts and adapt its strategy, resulting in better performance. In contrast, the bandit environment only provides feedback on the

chosen action, limiting the algorithm's ability to learn from mistakes and adjust its strategy. As a result, the cumulative regret of the Hedge algorithm in the bandit environment tends to be higher. By observing the mean regrets, it can be seen that the algorithm approaches the optimal mean regret while the horizon increases. This is because the algorithm has more data to learn from, which helps it make better predictions and reduce the impact of its mistakes.

Part 2 : In the second part of the exercise, the cumulative and mean regrets of UCB and hedge algorithm were compared in an adversarial bandit environment. The cumulative/mean regrets of both algorithms were plotted (*for horizons T = 1000 and T = 7000*) and analyzed to evaluate their performance.



Based on the plotted graphs, it can be observed that the UCB algorithm performs worse than the Multiplicative Weights (MW) algorithm in the adversarial bandit environment. The reason for this is that the UCB algorithm is designed for stochastic bandit problems (*such as the multi-armed bandit problem)*, where the rewards are stochastically generated, while the adversarial bandit environment in this exercise involves non-stochastic losses. In contrast, the MW algorithm is specifically designed for the adversarial environment, where the losses may be adversarially chosen. It is worth to note that as the horizon is increased, the regrets of UCB algorithm become smoother but they do not approach the optimal regret (*for the same reason described above*).