POLITECNICO
MILANO 1863

Online Learning Application
Academic Year 2021 - 2022

Pricing & Social Influence

Sofia Martellozzo Vlad Marian Cimpeanu Lorenzo Rossi
(cod persona) (cod persona) (cod persona)

Professor

Nicola Gatti

Contents

Introduction

Nowadays one big problem of e-commerces is to allocate the best price to its products so that, the seller can maximize its revenue.
The main issue is that increasing the price of a product leads to less people interested in that product, thus increasing the price is not necessarily beneficial to the seller. In contrast decreasing the price will increase the number of people interested in the product, but the revenue will be of course sub-optimal.
In order to maximize the revenue we can analyze the demand curve of a given product, which is a graphical representation of the relationship between the price p_i of a good or service i and the quantity demanded $q_i(p_i)$ for a given period of time, and find the price $\hat{p}$ such that:

$$\hat{p} = \arg\max_{p}(pq(p))$$

Unfortunately, in real world problems, the demand curve is not available, furthermore, we need to estimate this curve by interacting with the environment. One main problem of interacting with an unknown environment is that exploration costs a lot of money, so we want to find the best prices in the shortest amount of time to decrease the regret.
In order to do so, we can use reinforcement learning techniques such as Multi Armed Bandit (MAB) algorithms.

Practical example

In this project we want to study the case of a new e-commerce entering the market called ANS^2 that sells skateboarding clothes. More precisely, it is going to sell unisex t-shirts, hoodies, t-shirts, shoes and shirts.
For simplicity sake we can assume the website can sell an unlimited number of units without any storage cost whose goal is to minimize the cumulative regret while learning.
The web site of the vendor is structured as follows: in every webpage, a single product, called primary, is displayed together with its price. The user can add a number of units of this product to the cart. After the product has been added to the cart, two products, called secondary, are recommended. When displaying the secondary products, the price is hidden. Furthermore, the products are recommended in two slots, one above the other, thus providing more importance to the product displayed in the slot above. The website will propose only products that the customer has never seen before. If the user clicks on a secondary product, a new tab on the browser is opened and, in the loaded webpage, the clicked product is displayed as primary together with its price.

One main consideration we want to make is that the customer may buy different products during a visit, thus the price for a specific product may influence the total income generated by the customer.
For instance, let us assume that a customer lands on the webpage displaying a t-shirt: if the price is too high, the probability to buy that product is lower, but not only, also the probability to see the secondary products is lower, so it will decrease the probability that a customer visits and buys new products. In conclusion, when we choose the price for a specific product we have also to consider the indirect reward it will generate.

1 Step : Environment

2 Step : Optimization algorithm

Here we consider the case in wich all the parameters are known and the goal is to maximize the cumulative expected reward, following a greedy approach.
The algorithm works as follow:

1. set the prices of all the pruducts with the lowest one
2. collect the reward obtained by increasing, each time, the price of just one product of the original super arm
3. compare the five different configuration obtained with the first one. There could be two case
    (a) there is an increase of the reward, so select the one that gave the maximum reward (the highest increase) as the best one and repeat the algorithm from point 2
    (b) there is no increse (all the new configuration is worst than the previous one) and stop the algoritm
4. Return the actual best configuration.

For example:
The algorithm starts with the super arm with all the lowest prices for all the products : [00000]
Then explore by pulling the five combination of super arm, found by increasing of just one one product at time:
[00**1**00]
[0000**1**]
[000**1**0]
[**1**0000]
[0**1**000]
Now select the one with the highest reward, compare also with the originel best one. In our case it was: [00001]
Update the new super arm as the best one and start again, looking at the new five configuration available:
[0000**2**]
[00**1**0**1**]
[0**1**00**1**]
[**1**000**1**]
[000**11**]
And so on, until no improvement is found.

2.1 Limitations

This type of learner do not directly consider the parameters of a Customer, it just interacts with the environment by selecting the arm to pull at each round and observing the reward gibven by the environment.
Is guaranteed that the algorithm would not cycle, because it monotonically increases the prices (as well as the cumulative expected margin). On the other hand there is not guarantee that the algorithm will return the optimal price configuration.

2.2 Results

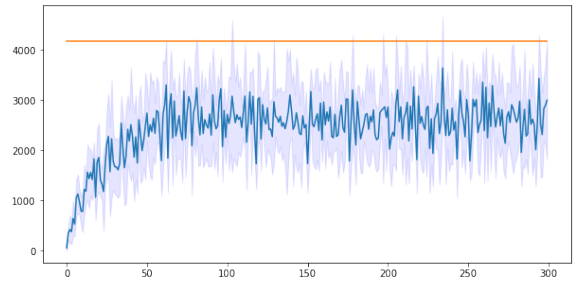


Figure 1: Reward

As said before here we can see that the algorithm converge to a solution that is not optimal,
so return a reward that is lower than the best possible one (clairvoyant)

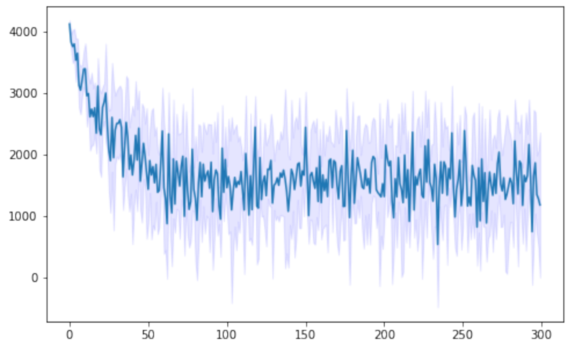


Figure 2: Regret

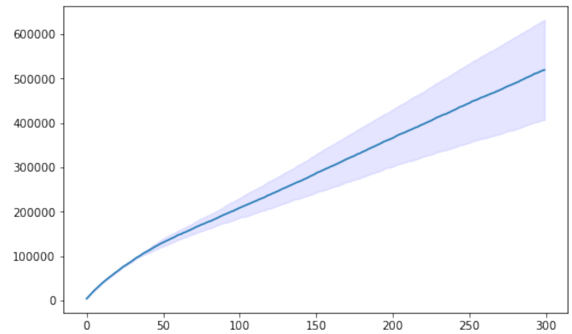


Figure 3: Cumulative regret

As expected the cumulative reward is linear (Figure 3)

3 Step : Optimization with uncertain conversion rates

3.1 Results

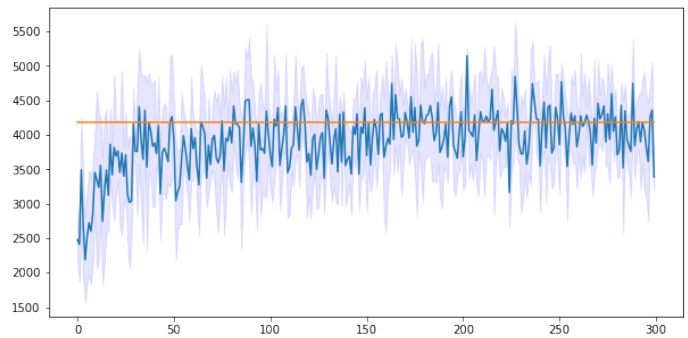


Figure 4: UCB Reward

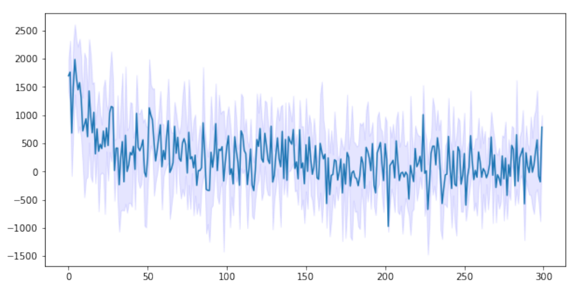


Figure 5: UCB Regret

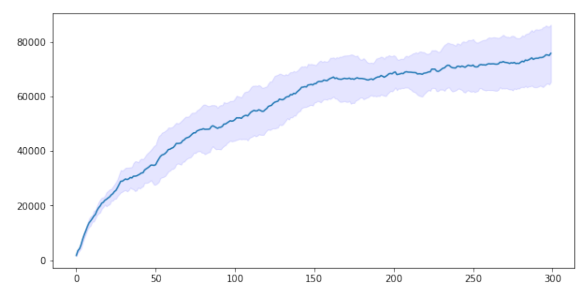
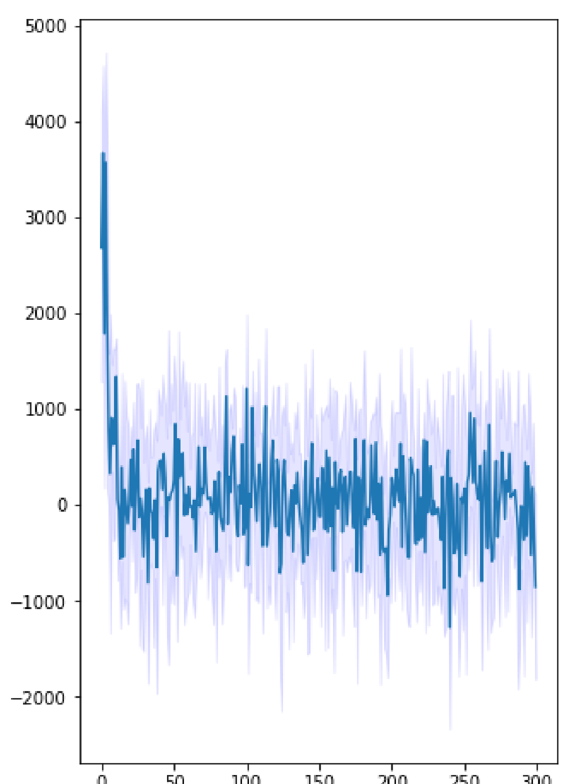


Figure 6: UCB Cumulative regret



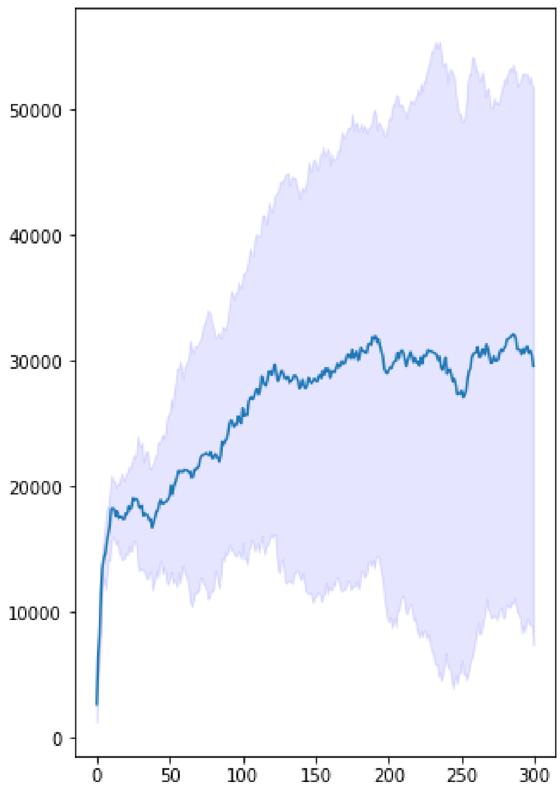


Figure 9: TS Cumulative regret

4 Step : Optimization with uncertain conversion rates, α ratios, and number of items sold per product

4.1 Results

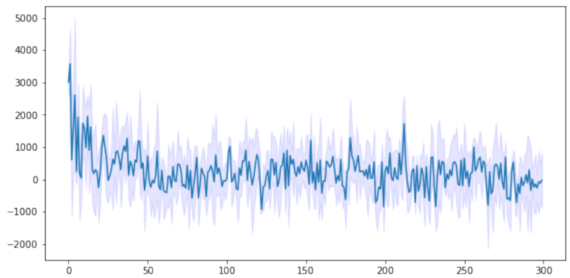


Figure 11: UCB Regret

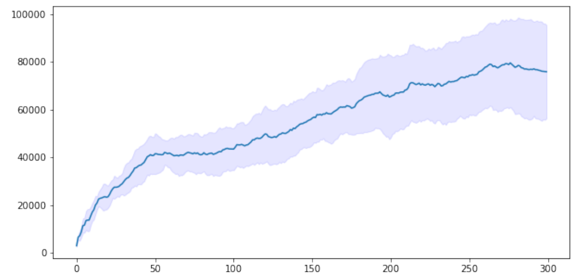


Figure 12: UCB Cumulative regret

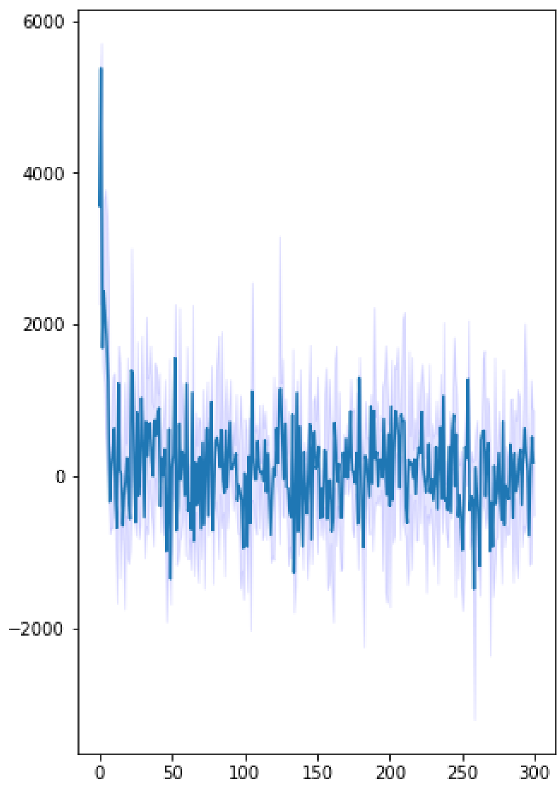


Figure 14: TS Regret

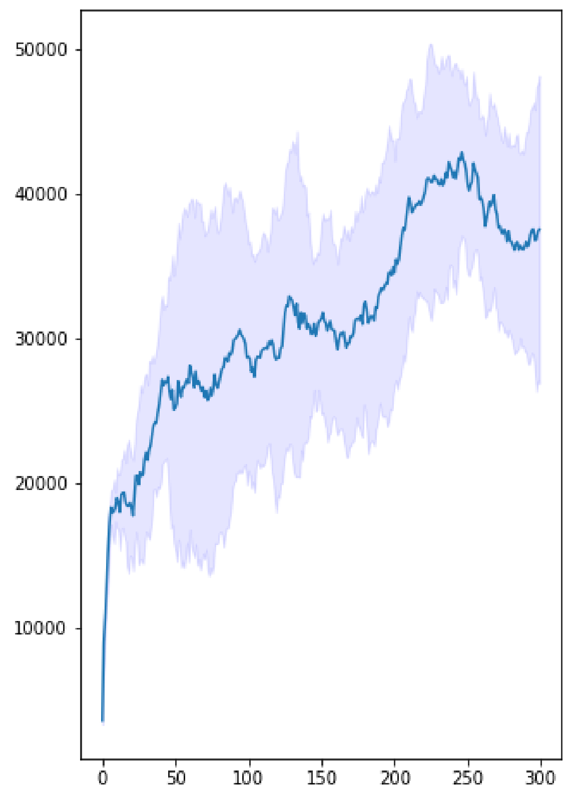


Figure 15: TS Cumulative regret

5 Step : Optimization with uncertain graph weights

5.1 Results

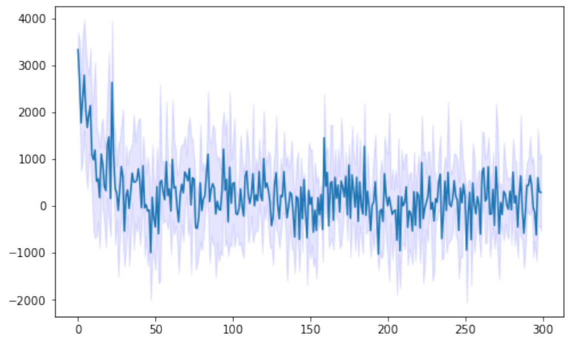


Figure 17: UCB Regret

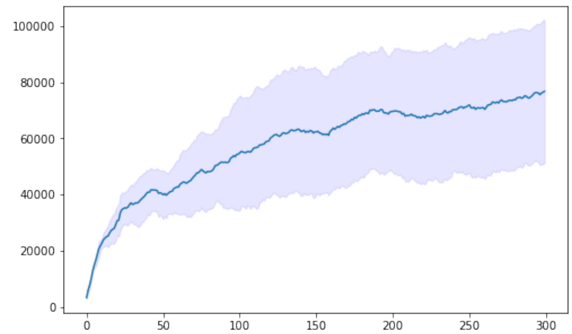


Figure 18: UCB Cumulative regret

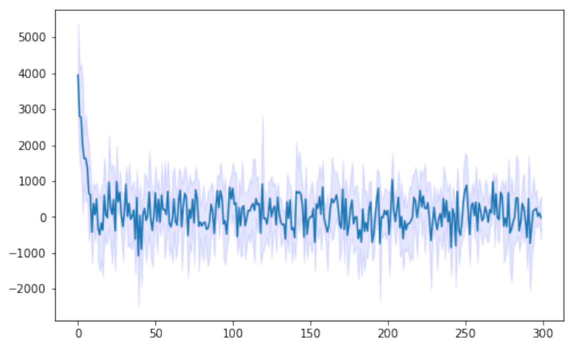


Figure 20: TS Regret

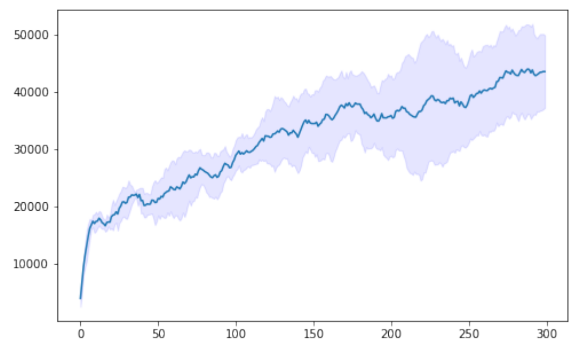


Figure 21: TS Cumulative regret

6 Step : Non-stationary demand curve

6.1 Results

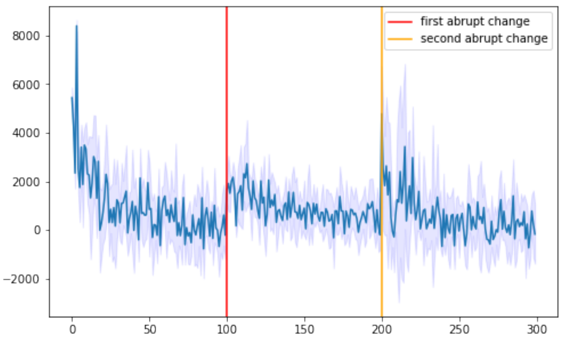


Figure 23: Change detection UCB Regret

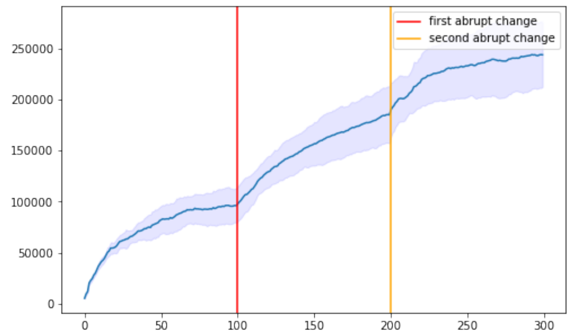


Figure 24: Change detection UCB Cumulative regret

7 Step : Context generation

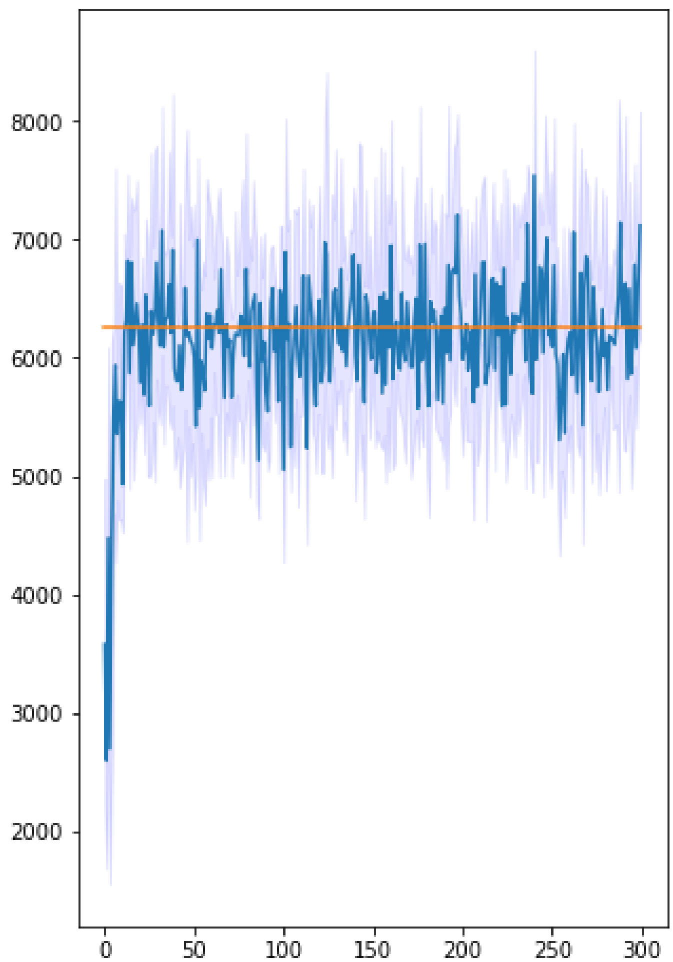


Figure 7: TS Reward

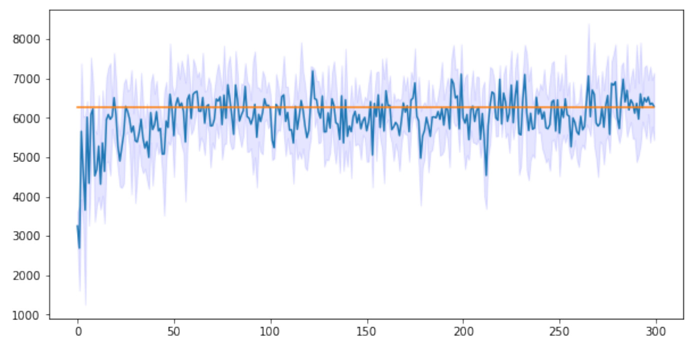


Figure 10: UCB Reward

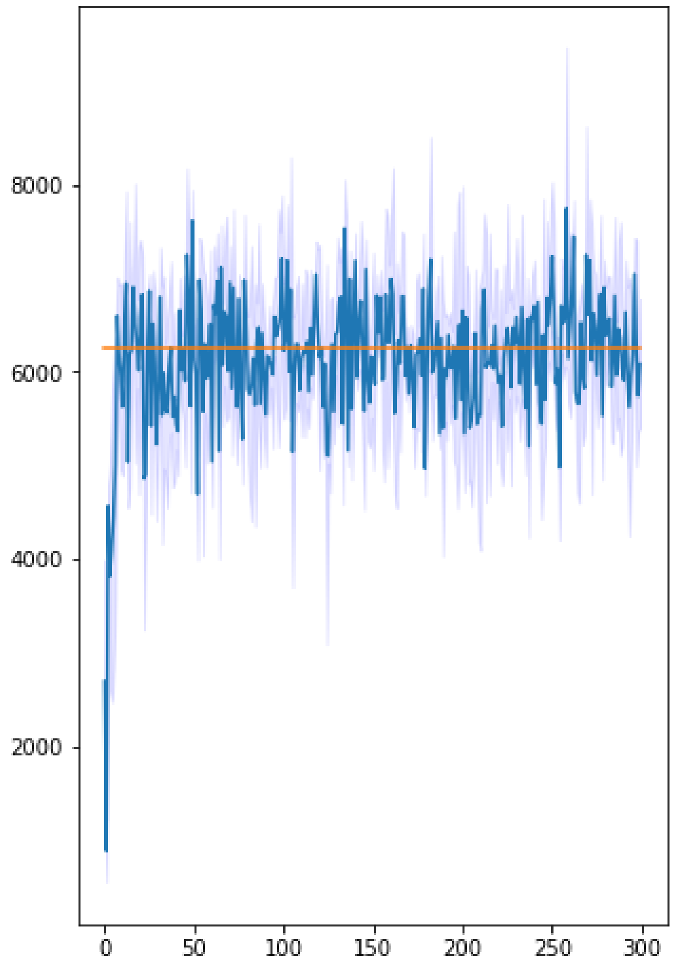


Figure 13: TS Reward

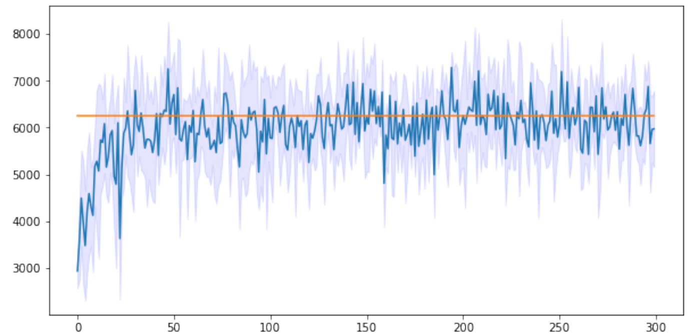


Figure 16: UCB Reward

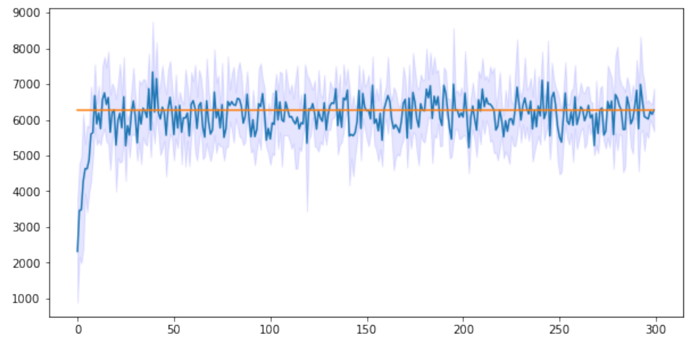


Figure 19: TS Reward

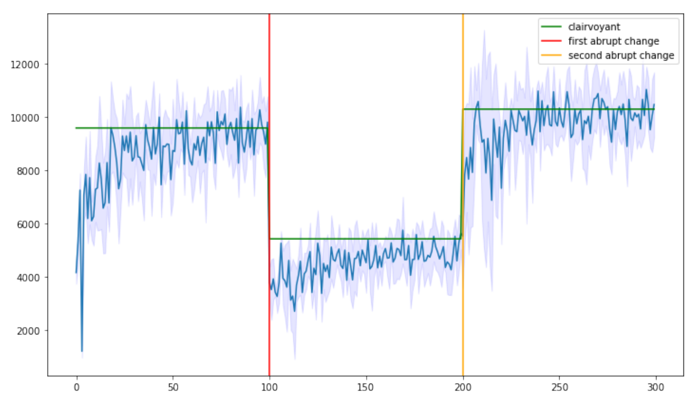


Figure 22: Change detection UCB Reward