In this technical project report, consider the situation where a store has many promotional codes for customers to purchase products so that customers can purchase a different product. Customers can belong to different categories, and promotional codes provide customers with different levels of discounts.

The essence of this problem is a multi-arm slot machine problem. We need to use algorithms to determine the price and discount distribution that will get the most profit. Among them, in terms of pricing, arm refers to the price of the product, and in terms of matching, arms is the best promotion task for each customer category。

**Scenes**

Consider a scenario where a retailer of technical consumer products sells two products. The first product is the iPad, denoted as product 1. The second product is iPad pencil, denoted as product 2. In order to increase sales of the iPad pencil, the retailer offers a promotional bundle to customers who purchase iPad. This means that all customers who purchase iPad will receive promotional offers for iPad pencil. The number of promotional bundles is preset by the retailer’s business department and provided to customers based on the customer category to which they belong. There are a total of four types of customers, and each customer is divided into one of the following four categories:

1.College students under the age of 1.25

(A) Willing to pay quite a lot for the iPad, but the budget is limited

(B) If you are interested in buying iPad pencil, if the product functions are reasonable, they will pay a higher price.

2. White-collar workers over 30 years old.

(A) Willing to pay a higher amount for the iPad and have a larger budget limit

(B) Not much interest in iPad pencil, but if the price is attractive, they will buy

3. Old customers over 50 years old

(A) The interest in buying technology products is not as good as the first two categories, so they are unwilling to spend so much money

b) The interest in iPad pencil is not as good as iPad, but if there is a discount, they will buy it.

4. "Fruit Powder"

(A) As fans, they expect attractive prices, but they have a trusting relationship with retailers, so they have an incentive to buy iPads from retailers

(B) Due to the bundled products provided by retailers, consumers are encouraged to buy iPad pencil, and as the discounts on bundled products increase, consumer interest also increases

The retailer’s iPad price is $650 per unit. The following table shows the conversion rate for each category of the seven candidate prices.

chart1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  | 700/50 | 800/150 | 900/250 | 1000/350 | 1100/450 | 1200/550 | 1300/650 |
| Class 1 | 0.50 | 0.65 | 0.55 | 0.52 | 0.35 | 0.16 | 0.05 |
| Class 2 | 0.55 | 0.60 | 0.50 | 0.45 | 0.40 | 0.35 | 0.20 |
| Class 3 | 0.45 | 0.40 | 0.35 | 0.25 | 0.15 | 0.10 | 0.05 |
| Class 4 | 0.65 | 0.75 | 0.66 | 0.55 | 0.35 | 0.25 | 0.15 |

Consider a customer buying an iPad, and then giving the customer a promotion. The promotion includes four different discount levels on the iPad pencil. The four levels are P0, P1, P2, and P3. They give discounts of 0%, 10%, 20% and 25% respectively. For retailers, the cost of the iPad pencil is $50 each. The following table lists the price and profit of each original price and promotion.

chart2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  | Price1 | Price2 | Price3 | Price4 | Price5 | Price6 | Price7 |
| P0 | 80/30 | 90/40 | 100/50 | 110/60 | 120/70 | 130/80 | 140/90 |
| P1 | 72/22 | 81/31 | 90/40 | 99/49 | 108/58 | 117/67 | 126/76 |
| P2 | 64/14 | 72/22 | 80/30 | 88/38 | 96/46 | 104/54 | 112/62 |
| P3 | 60/10 | 67.5/17.5 | 75/25 | 82.5/32.5 | 90/40 | 97.5/47.5 | 105/55 |

Table 3 shows the conversion rate of all categories for a given price and promotion：

chart3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Price1 | Price2 | Price3 | Price4 | Price5 | Price6 | Price7 |
| Class1/P0 | 0.55 | 0.53 | 0.52 | 0.47 | 0.46 | 0.42 | 0.35 |
| Class1/P1 | 0.62 | 0.59 | 0.56 | 0.54 | 0.53 | 0.50 | 0.48 |
| Class1/P2 | 0.65 | 0.63 | 0.60 | 0.59 | 0.58 | 0.56 | 0.55 |
| Class1/P3 | 0.70 | 0.68 | 0.65 | 0.62 | 0.58 | 0.56 | 0.53 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class2/P0 | 0.35 | 0.33 | 0.32 | 0.27 | 0.24 | 0.22 | 0.15 |
| Class2/P1 | 0.42 | 0.39 | 0.36 | 0.34 | 0.32 | 0.30 | 0.27 |
| Class2/P2 | 0.55 | 0.50 | 0.43 | 0.39 | 0.35 | 0.33 | 0.29 |
| Class2/P3 | 0.50 | 0.48 | 0.45 | 0.42 | 0.38 | 0.36 | 0.33 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class3/P0 | 0.15 | 0.13 | 0.12 | 0.07 | 0.06 | 0.04 | 0.03 |
| Class3/P1 | 0.22 | 0.19 | 0.16 | 0.14 | 0.10 | 0.07 | 0.04 |
| Class3/P2 | 0.35 | 0.33 | 0.27 | 0.26 | 0.23 | 0.16 | 0.10 |
| Class3/P3 | 0.39 | 0.36 | 0.33 | 0.29 | 0.25 | 0.19 | 0.13 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class4/P0 | 0.50 | 0.38 | 0.32 | 0.30 | 0.26 | 0.22 | 0.15 |
| Class4/P1 | 0.52 | 0.49 | 0.46 | 0.35 | 0.29 | 0.20 | 0.18 |
| Class4/P2 | 0.65 | 0.53 | 0.50 | 0.49 | 0.45 | 0.36 | 0.25 |
| Class4/P3 | 0.70 | 0.68 | 0.56 | 0.52 | 0.48 | 0.46 | 0.43 |

Make the following assumptions in this scenario：

•On average, 1,000 customers arrive at the store every day

•The average number of customers in each category are:

Distribution of categories 1, 2, 3 and 4 [400, 300, 100, 200]

• Promotional distribution settings: [0.3, 0.15, 0.25] are respectively for promotion levels P1, P2 and P3.

P0 is unrestricted in all cases because it corresponds to no discount in all cases

• Conversion rate of non-stationary second stage:

– The conversion rate of the first product drops in category 1&.2, while it remains unchanged in category 3&.4

-The conversion rate of the second product dropped in category 3&.4, but remained unchanged in category1&.2

**Task 1.Mathematical formula**

Parameters indicate:

：Is the average number of customers of this category who arrive at the store every day.

：Is the price selected by the store for product 1 at time t.

:Is the price selected by the store for product 2 at time t.

:Refers to the profit when the store sells a product 1 at price P.

：Refers to the profit when the store sells a product 2 at price Q and promotion F.

：Is the conversion rate of type i of the first product at time t, assuming the price is P.

:Is the conversion rate of category i of the second product at time T, taking into account price Q and promotion F.

Is the score of promotion f provided to category i at time t.

For the first product, the total revenue of the store in a day will be equal to the price profit it chooses multiplied by the number of customers purchased, which is equivalent to the average number of customers in each category multiplied by the conversion rate of the category at the design price with. This means that the total profit can be expressed as：



The number of customers of each type who can purchase the second product only corresponds to the part of the first product, and then the total number of customers who purchase the second product is：



We need to aggregate all promotions to get the total amount. If we add up all categories and include the profit of the second product, we will get the second part of the total daily profit of the store, corresponding to：



In a one-year time interval, by adding the two total profit items, we get the total revenue of the store in one year：



This is the amount the store is trying to maximize, and the variables they need to modify are the price and quantity of the two products and the price and quantity of promotions offered for each category. Then the store needs to solve the following optimization problems：





The only restriction we need to add is that the sum score of the promotions offered is equal to 1. The nature of this optimization problem depends entirely on the functions sm, N, RandS. If the function is linear, then the whole problem becomes a linear optimization problem, which can be solved using the classic simplex algorithm. If the function is not linear, then we need to use a nonlinear solver to solve it. For example, if the function is continuous, any method derived from the gradient descent algorithm will provide a suitable solution. Finally, if prices and scores can only belong to a finite set, then the nature of the function becomes irrelevant, but the problem is transformed into a combinatorial problem, which needs to be solved by brute force methods.

Finally, since the parameters and variables of one day will not affect the other days, the original problem can be broken down into small problems, and then concentrated in the daily optimization.

**Task 2 Uncertainty mathematical formula**

In this section, we need to consider three different sources of randomness. First, the number of various types of customers that arrive at the store every day, then how many of these customers will buy the first product, and finally how many of the customers who buy the first product will buy the second product.

The number of customers arriving each day can be modeled by supporting any distribution of natural numbers. We will represent the distribution of the number of customers in Ci, t(θi, t), where θi, tare weight are the corresponding distribution parameters.

The number of customers who actually bought the first product can be modeled as the result of the Bernoulli experiment, where the number of experiments is the number of customers that arrive, and the probability of success is the category conversion rate calculated at the corresponding price. Then, the number of buyers for the first product can be modeled as a binomial distribution, with the number of trials equal to the number of customers, and the probability of success equal to the conversion rate.

Finally, the sales volume of the second product can also be modeled as a Bernoulli test, but in this case, the number of trials corresponds to the number of buyers of the first product, and the probability of success is the conversion rate of the second product . In addition, we need to consider where the customer’s promotion is provided.

Then, the deterministic optimization problem in Part 1 can be modified to the following stochastic optimization problem:











In the formula,Ci，t（θ） is any probability distribution that supports natural numbers and parameters（θ）, and B (c, p) is a binomial distribution without the number of trials and the probability of success. In this case,ci，t，χ1i，tandχ2i, t are a random variable, modeling the number of customers who arrive at the store in one day, the number of people who bought the first product, and the number of people who bought the second product.

Since this optimization problem is not deterministic, we cannot use the same method as in the previous section, we need to use online learning methods. For this kind of problem, we can use the multi-arm bandit method, where each arm is a combination of the prices of two products and the effective distribution of promotional activities, and the parameters are updated every day.

**Task 3**

**3.1 problem**

The goal of this section is to provide a solution to the pricing problem for product 1 iPad. In the solution of learning the best price of the first product, the price of the second product iPad pencil is fixed, which means that during the learning process, the promotion tasks and prices are fixed. The known problem parameters in this section are the conversion rate of product 2 and the number of customers arriving each day and their category distribution. The daily customers are extracted from the Gaussian distribution. Since the number of customers is known, the number of rounds per day is known.

## 3.2. Method

Use the upper limit of confidence method (UCB) and Thompson sampling method to learn the optimal price of the first item and compare its performance. In this method, the number of ARMs is equal to the number of candidate prices for product 1, a total of seven. The number of customers arriving at the store every day is set to Classs i={400,300,100,200}. The conversion rate of product 2 is the same for each ARM. Using a static environment, two learners are used to simulate the learning process for 365 days a day. Every day, the arms of two independent learners will be pulled, the reward of the selected arm will be calculated, and the learner will be updated.

**3.3 Results**

Both Thompson sampling and UCB converge to 1000.

LEARNING RESULTS

Thompson learner converges to price 1000 for product 1

UCB learner converges to price 1000 for product 1

### Reward

Total margin collected by UCB: 278.7577736203766

Total margin collected by Thompson Sampling: 326.90383816625774

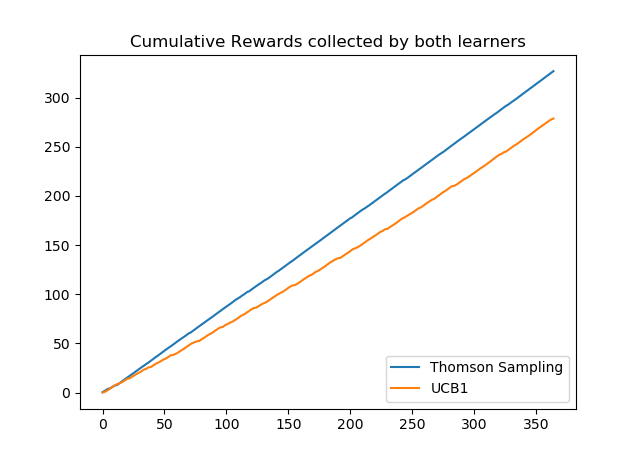


Figure-1

Figure 1 shows the cumulative rewards collected by these two algorithms. The cumulative reward collected by the Thompson sampling algorithm is slightly larger than that of the UCB algorithm.

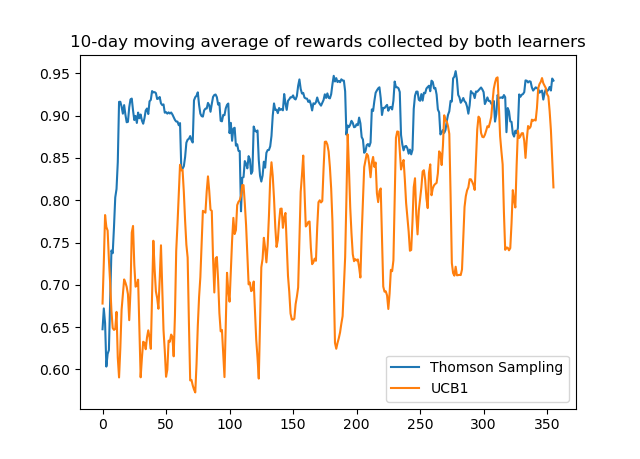


Figure-2

Figure 2 shows the 10-day moving average of the rewards received by two learners. In this figure, it can be observed that the Thompson sampling algorithm reaches a higher 10-day moving average faster, and the change after reaching the higher 10-day moving average is smaller than UCB. The 10-day moving average of the UCB algorithm takes longer to reach the value of the Thompson algorithm, and the change of UCB is larger than that of the Thompson algorithm.

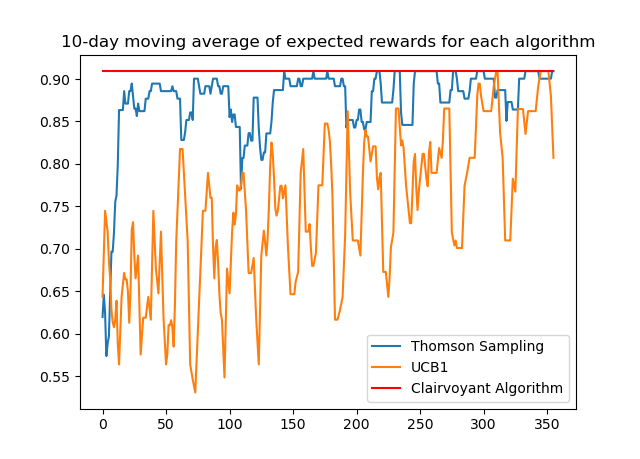


Figure-3

Figure 3 shows the 10-day moving average of the expected reward for each algorithm. These three algorithms are Thompson sampling algorithm, UCB algorithm and perspective algorithm. The perspective algorithm is the optimal solution of the problem calculated when all the parameters in the problem are known. Studying the graph in Figure 3, we can find that the solution produced by Thompson's sampling algorithm is closer to the optimal solution.

### Regrets

Total expected regret of UCB: 61.74371428571428

Total expected regret of Thompson Sampling: 14.00110714285714

## **Task** 4

## 4.1 problem

Similar to part 3, but in this case, the conversion rate associated with product 2 and the number of daily customers are unknown. The goal is to provide solutions to pricing problems for products 1 and 2. The number of unknown daily customers comes from a Gaussian distribution. Apply the learning process to each arriving customer every day, and randomly select the customer categories that arrive at the customer according to the predefined category distribution.

4.2 Method

In order to understand the optimal price of the first product and the second product, the upper limit of confidence method (UCB) is used. Unlike Part 3, the second product introduces 7 candidate prices. Like task 3, the product promotion task is fixed. The number of arms among the learners of Product 1 and Product 2 will be 7 respectively, which is equal to the number of candidate prices. In the learning process, the number of customers in each category is sampled from a normal distribution. In the simulation, the use of customer arrivals is a randomly selected class in which there are customers remaining on the day. Pull the arm of product 1 and observe the reward. If the reward of product 1 is positive, that is, the customer purchases the first product, pull the arm of the second product to calculate the reward. Then update the two learners with the corresponding arm and calculate the cumulative reward.

4.3 **Results**

LEARNING RESULTS

learner 1 converges to price 1000 for product 1

learner 2 converges to price 100 for product 2

### For constant matching {(Class 1, P0), (Class 2, P1), (Class 3, P3), (Class 4, P2)}, the learner of product 1 converges to 1000, and the learner of product 2 converges to 100 .

### **Reward**

Total profit collected from product 1: 60709650.0

Total profit collected from product 2: 3447140.0

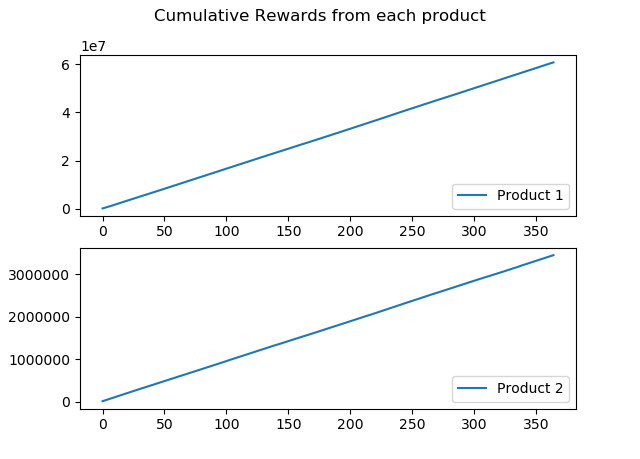


Figure-4

Figure 4 shows the cumulative reward for each product in the two product pricing problems.

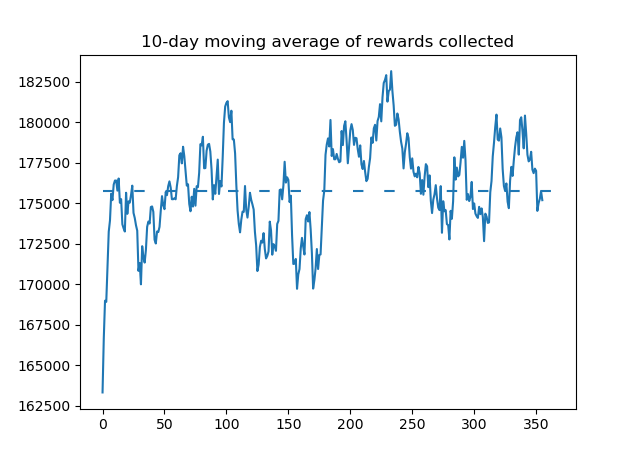


Figure-5

Figure 5 shows the 10-day moving average of the rewards collected by the UCB algorithm

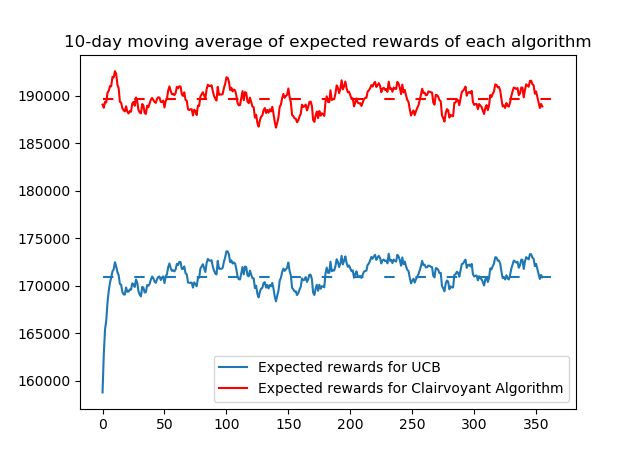


Figure-6

### Figure 6 shows the 10-day moving average of the expected returns of the UCB- and perspective algorithms. The UCB algorithm has never achieved the high average expected return of the clairvoyance algorithm. It can be seen from the figure that the average value of UCB in 365 days is about 86,000, while the average value of clairvoyance is about 92,000, which means that UCB has reached 89% of the moving average of the expected reward of the clairvoyance algorithm in 10 days.

### regret

Total expected regret: 6795174.360501035

# task5

## 5.1.problem

## This section focuses on matching issues. More precisely, the goal is to find the best distribution of promotions for each customer category. To simplify the problem, the price is fixed. At this point, these scores have two optional settings, as described in the scenario section.

## 5.2.method

## The method is very similar to the previous part. The main difference is that arms are no longer price candidates, but the best promotional tasks for each customer category. This boils down to the use of an allocation algorithm on the matrix, where the rows represent customer categories and the columns represent promotions. Since each category can be assigned to the first promotion, three additional promotion categories are introduced.

## For each day, we will determine the number of promotional activities based on the number of customers. Once the arm of product 1 is pulled and the observed reward is positive, we solve the matching problem. If there are no more promotions for certain types, the cost of this column in the matrix will be as high as possible, so this column will never be selected. After the matching problem is resolved, we determine which promotion has been selected for the arriving customer. The learner will be updated with the observed reward, and the number of daily promotions for matching types will be reduced.

## 5.3**Results**

LEARNING RESULTS

The number of times a class was offered each promo level is shown below:

[[5.0700e+02 2.7230e+03 5.9346e+04 1.0198e+04]

[5.2394e+04 0.0000e+00 7.6350e+03 5.0000e+00]

[1.7900e+02 1.0000e+00 0.0000e+00 1.6146e+04]

[2.0600e+02 4.7177e+04 0.0000e+00 1.0000e+01]]

### reward

Total profit collected from product 1: 9826350.0

Total profit collected from product 2: 1880806.0

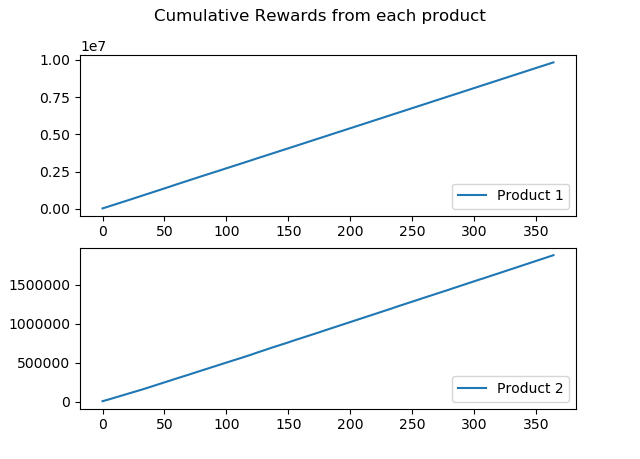


Figure-7

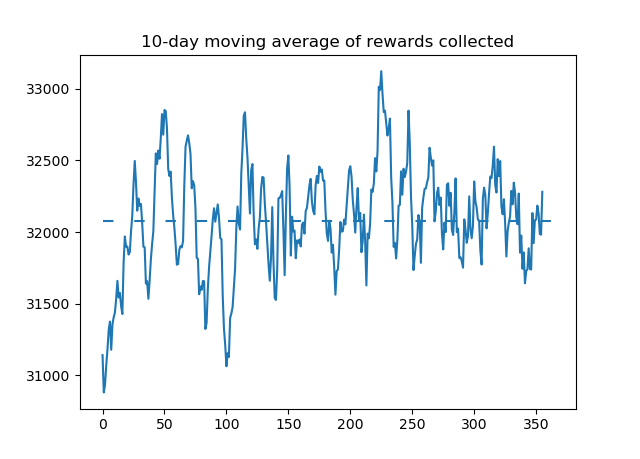
Figure 7 shows the cumulative reward.

Figure-8

Figure 8 shows the 10-day moving average of the rewards collected by the UCB algorithm

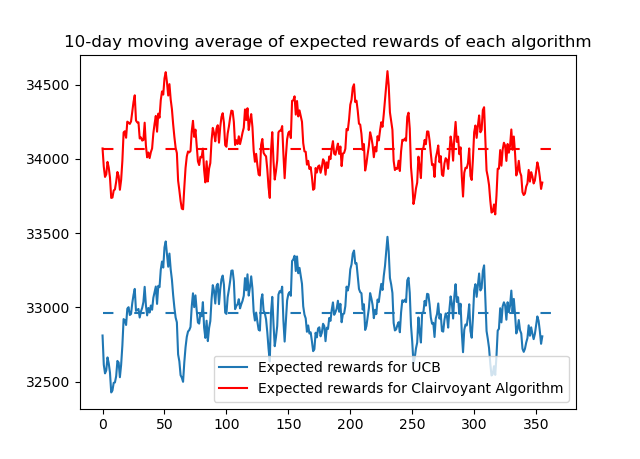


Figure-9

Observe the difference between the 365-day average of UCB and fluoroscopy in Figure 9. The average value of UCB is 14 400, and the average value of clairvoyance is 14 900, which means that UCB has reached 97% of the 10-day moving average of the expected rewards of the clairvoyance algorithm throughout the year. This is an improvement on the previous point 4.

### regret

Total expected regret: 403812.88000002666

# task6

## 6.1problem

Similar to Part 5, except that the pricing of product 1 and product 2 is no longer fixed. The problem now is pricing and matching. In these sections, the following assumptions are made about the environment: it is fixed and the arrival of customers is continuous, which means that the rewards returned for each customer are based on level and price, rather than rewards throughout the day. The number of customers per day comes from a Gaussian distribution. Apply a learning process for each arriving customer every day. According to a specific category distribution, the customer category to which the arriving customer belongs is randomly selected. The promotion level distribution is set to a small fraction of the total customers that arrive and is constant. There are two optional settings for these scores, as described in the scenario section.

## 6.2method

The methods used in this section are the UCB method for the first product pricing problem and the matching UCB method for the second product matching problem. The number of arms of the pricing learner is equal to the number of candidate prices, and there are a total of 7 candidate prices. For the second matching problem of product 2, the number of arms should be equal to 112, but in order to use the linear sum allocation algorithm to solve our optimization problem, another 84 arms were created. These weapons are additional copies of the 0 upgrade mission. As before, the number of customers in each category is sampled from a normal distribution and truncated to 0 to avoid negative numbers. The simulation of customer arrival is carried out by randomly selecting categories with remaining customers on the day. As in the previous part, we pull the arms for the first product and observe the rewards. If the reward is positive, we will pull the second price arm, otherwise the reward for the second product is set to zero. Then update the learner. For each customer, the rewards calculated for each customer are aggregated into cumulative rewards, and the expected rewards for each day are calculated from this.

## 6.3**Results**

The result of the promotion setting. When level 1 students are at promotion level 1, the price of product 1 converges to 900, and the price of product 2 converges to 70. When level 2 students are at promotion level 2, the price of product 1 converges to 1100, and the price of product 2 Convergent to 70, at promotion level 3, the price of category 3 products is 800, and the price of category 2 products is 70; at promotion level 0, the price of category 4 products is 800, and the price of category 2 products is 75. In the two tables below, the number of times each promotion level is assigned to a class is given.

LEARNING RESULTS

class 1 learners converged to price 900 for product 1 and price 80 for 2

class 2 learners converged to price 1100 for product 1 and price 80 for 2

class 4 learners converged to price 900 for product 1 and price 90 for 2

With the following shows the number of times a class was assigned each promo level:

[[2.0000e+00 3.9535e+04 4.3000e+01 6.8000e+01]

[2.0000e+00 0.0000e+00 2.2217e+04 4.9000e+01]

[1.0000e+00 0.0000e+00 0.0000e+00 5.0200e+03]

[2.0000e+00 2.4101e+04 0.0000e+00 1.0000e+00]]

### reward

Total profit collected from product 1: 26675550.0

Total profit collected from product 2: 1096076.5

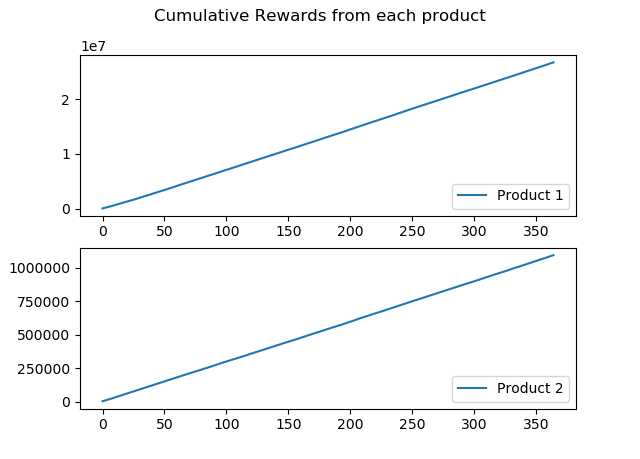


Figure-10

Figure 10 shows the cumulative reward for each product.

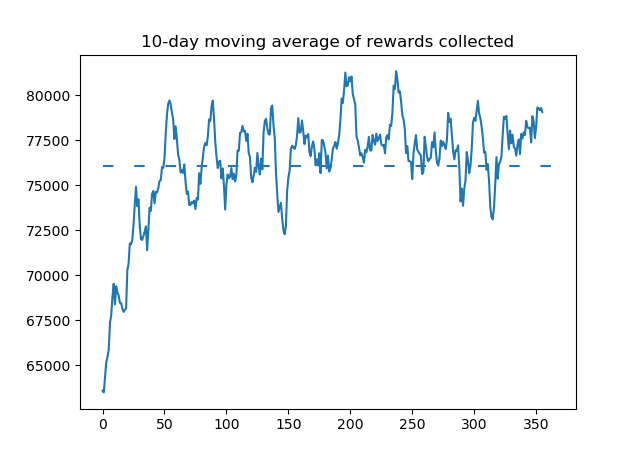


Figure-11

Figure 11 shows the 10-day moving average of the collected rewards.

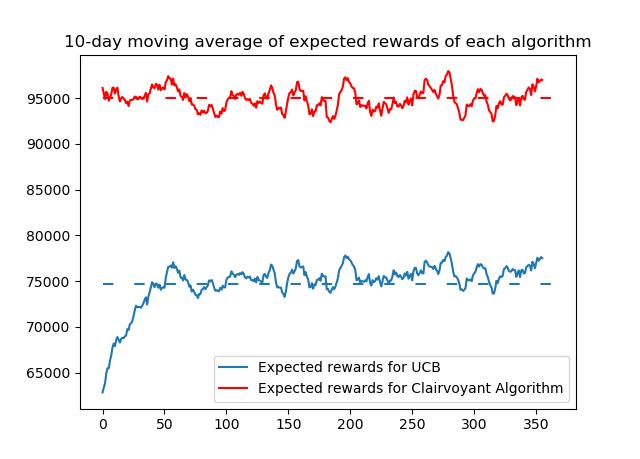


Figure-12

### Figure 12 shows the 10-day moving average expected rewards of UCB and the perspective algorithm.

### regret

Total expected regret: 7394228.551199923

# Task7

## 7.1problem

## The scope of this section is to test the algorithm in a non-stationary environment, which means that the parameters will not remain constant over time. In our example, we divide the year into two phases of equal time: in the first phase, all parameters are the same as those in the previous task; in the second phase, the conversion rate is in accordance with the The description is changed.

## 7.2method

## As the parameters are constantly changing, learners need to be able to adapt to these changes. One method is to use the sliding window method. In this method, we choose a time window of length and only use the last day's return to train the learner. This technique allows learners to update parameters and ignore information that may be out of date. In order to adapt the algorithm to our approach and ensure that all customers are considered in a round, we need to use a window of length equal to the square root of the period times the number of expected customers in a day.

## 7.3**Results**

LEARNING RESULTS RESULTS

Phase 1 results:

class 1 learners converged to price 1000 for product 1 and price 80 for 2

class 2 learners converged to price 1200 for product 1 and price 80 for 2

class 3 learners converged to price 900 for product 1 and price 80 for 2

class 4 learners converged to price 1000 for product 1 and price 90 for 2

With the following shows the number of times a class was assigned each promo level:

[[1.0000e+01 1.7866e+04 1.2000e+01 0.0000e+00]

[1.0000e+01 6.0000e+00 1.1115e+04 0.0000e+00]

[1.0000e+01 0.0000e+00 1.5000e+01 2.3090e+03]

[2.0000e+01 9.9270e+03 0.0000e+00 1.1700e+02]]

Phase 2 results:

class 1 learners converged to price 1000 for product 1 and price 80 for 2

class 2 learners converged to price 1100 for product 1 and price 80 for 2

class 3 learners converged to price 900 for product 1 and price 80 for 2

With the following shows the number of times a class was assigned each promo level:

[[1.0000e+01 1.1486e+04 0.0000e+00 0.0000e+00]

[1.0000e+01 8.0000e+00 6.8050e+03 0.0000e+00]

[9.0000e+00 0.0000e+00 1.2000e+01 2.2480e+03]

[1.9000e+01 9.6850e+03 0.0000e+00 8.1000e+01]]

### reward

Total profit collected from product 1: 21526900.0

Total profit collected from product 2: 790579.0

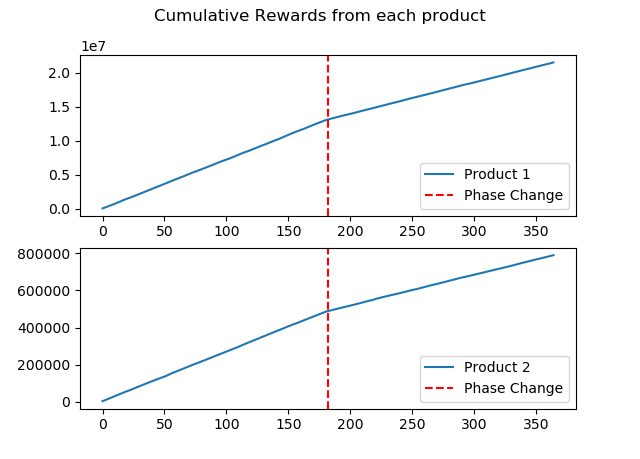


Figure-13

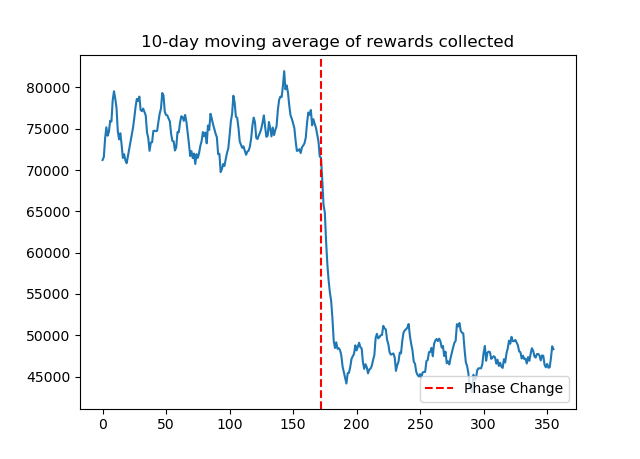
In Figure 13, the effect of non-stationarity can be observed. Specifically, we can observe that the cumulative reward slope of product 2 is smaller in the second stage than in the first stage. This decrease is due to the reduced conversion rate of the product, which translates into a decrease in sales.

Figure-14

We can see that one of the weaknesses of the UCB method is that it requires a lot of time and information to converge to the optimal configuration, especially in situations like our problem where the number of weapons to be tested is high. Since we always forget information, the sliding window approach adds to this weakness. This is shown in Figure 14. We can see that the daily reward does not increase, but oscillates between the average value of each stage.

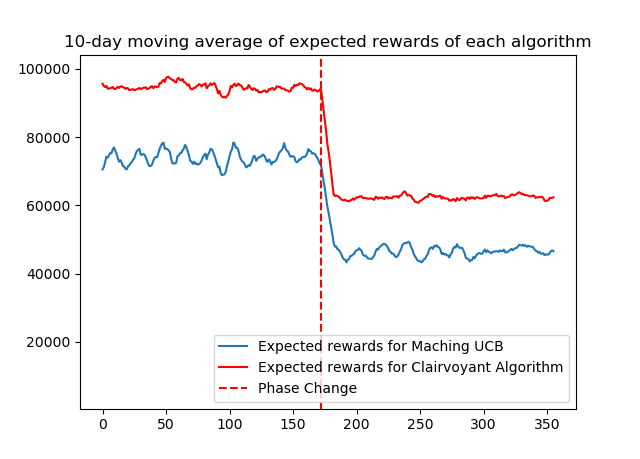


Figure-15

### When we compare the results of the perspective algorithm with our results, we can clearly prove this phenomenon. As shown in Figure 15, we can see that the return of the UCB algorithm will never be close to the optimal result.

### regret

Total expected regret: 6691581.237967253

# Task8

## 8．1problem

## The goal of this section is to test the algorithm in a non-stationary environment again, but now the change detection method is used instead of the sliding window method. Therefore, the parameters will not remain the same over time. The environment settings are the same as in the previous section.

## 8.2method

## This method uses the Cumulative Sum (CUSUM) algorithm, which has been proven to be the best algorithm for detecting mutations. The basic idea of the CUSUM algorithm is to use the function of the observed sample as the step size of the random walk. This random walk is designed to have a positive average drift after a change point, and a negative average drift when there is no change. Therefore, if the random walk exceeds a certain positive threshold, CUSUM will send a change signal.

## 8. 3**Results**

LEARNING RESULTS

Phase 1 results:

class 1 learners converged to price 1000 for product 1 and price 80 for 2

class 2 learners converged to price 1200 for product 1 and price 80 for 2

class 3 learners converged to price 900 for product 1 and price 80 for 2

class 4 learners converged to price 1000 for product 1 and price 90 for 2

With the following shows the number of times a class was assigned each promo level:

[[1.0000e+00 1.8693e+04 3.9000e+01 1.1000e+01]

[1.0000e+00 0.0000e+00 1.0065e+04 1.5000e+01]

[1.0000e+00 0.0000e+00 0.0000e+00 2.4790e+03]

[2.0000e+00 1.0247e+04 1.9000e+01 0.0000e+00]]

Phase 2 results:

class 1 learners converged to price 1000 for product 1 and price 80 for 2

class 2 learners converged to price 1000 for product 1 and price 80 for 2

class 3 learners converged to price 900 for product 1 and price 80 for 2

class 4 learners converged to price 1000 for product 1 and price 90 for 2

With the following shows the number of times a class was assigned each promo level:

[[ 0. 11810. 0. 0.]

[ 0. 0. 6938. 0.]

[ 0. 0. 0. 2617.]

[ 0. 10224. 0. 0.]]

### reward

Total profit collected from product 1: 25617200.0

Total profit collected from product 2: 806205.0

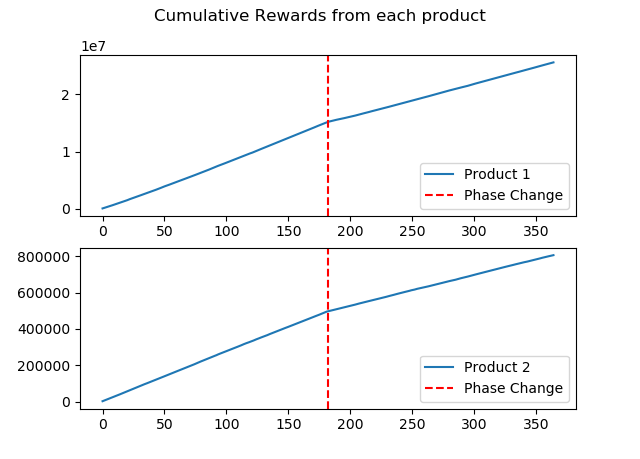


Figure-16

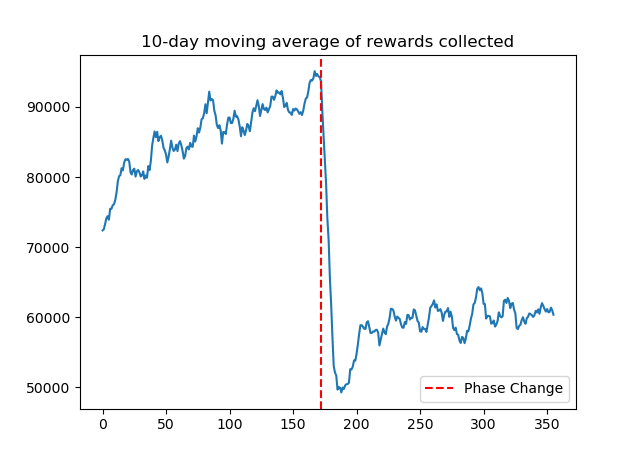
The behavior observed in Figure 16 is similar to that of Figure 13. Since the conversion rate of these two products is reduced, the cumulative reward slope of these two products is smaller in the second stage

Figure-17

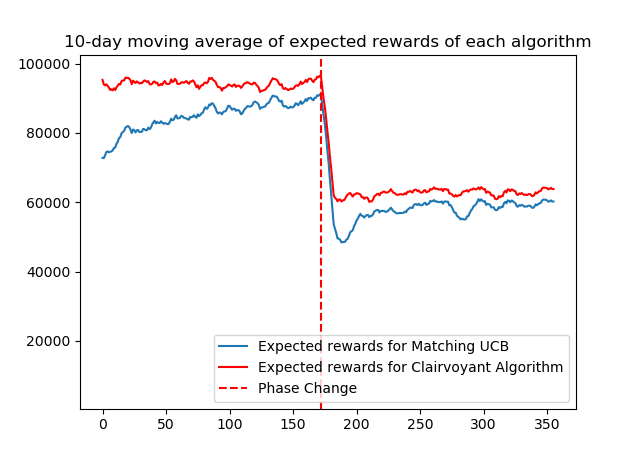
In Figure 17, we can clearly see that the graph is slowly converging to a certain value. Once the second phase begins, the algorithm detects sudden changes in the environment and restarts the convergence process.

Figure-18

### As shown in Figure 18, we can see that the return of the UCB algorithm is improved compared to the sliding window, and it is closer to the best result.

### regret

Total expected regret: 2635924.3168814387