Optimization algorithm

The purpose of this environment is to allow us to evaluate the ecommerce website. We recall that this website proposes five products to the client and the goal of the pricing proposal is to optimize the price of every product.

To do so, each product takes its price value in a set of four values, ordered increasingly, and we must choose the price that leads to the greatest income for the website. In other words, we must optimize the cumulative expected margin over all the products. We assume that the set of fourth values is directly the margin of this price for the product. The optimization problem can be state as follow :

$$max_{X_{j,t}} \sum_{j=1}^{5} X_{j,t} * n_j(X_{j,t})$$

where :

- $X_{j,t}$ is the margin value of the product j at time t. So :$X_{j,t} \in \{x_1, x_2, x_3, x_4,\}$

- $n_j(X_{j,t})$ represent the number of expected sales for the product j given its margin value at time t

We can clearly see that this last variable of the problem needs to be estimated. In fact it can be written as :

$$n_j(X_{j,t}) = CV(X_{j,t}) * nbUser * E(A) * P(reach\ webpage\ of\ X_j)$$

where :

- $CV(X_{j,t}) \in [0, 1]$ is the conversion rate for the given margin (equivalent to price we recall), times the number of user gives the number of user that will buy the product

- A is the random variable explaining how many product the user in question will buy the item (this is user dependant only)

- $P(reach\ webpage\ of\ X_j) \in [0, 1]$ is explicit its implicitly expression can be written as $P(reach\ webpage\ of\ X_j) = \alpha_j + \sum_{i=1 \neq j}^{5} \alpha_i * (\prod graph\ weight\ to\ reach\ X_j)$. To say it easily, it is just the sum of all the way to reach the product, so depend of the alpha ratio and the graph weights

Now that we have the full expression of the problem, we can start to solve it. In step 2, we assume that all parameters that are involved in the equation are known. Thus, we have for each product and for each margin of it, the number of expected sales.

The greedy algorithm used is defined as : at the beginning, every item is associated with the corresponding lowest price. Then, evaluate the marginal increase obtained when the price of a single product is increased by a single level, thus considering 5 potential different price configurations at every iteration, and choose the price configuration providing the best marginal increase (a price configuration specifies the price of every product). The algorithm stops when no new configuration among the 5 evaluated is better than the previous one.

This algorithm is not optimal and may not return the best configuration price. In fact, if for one product, the product margin times number of expected sales of the next margin value is lower than the same multiplication with margin two position after, this algorithm will not detect this and so don't learn the optimal choice.

As well, if the best choice is to set all the margin to the last (4th), and the algorithm could learn it, meaning $X_{j,t} * n_j(X_{j,t})$ is increasing for each margin value and each product, this is the worst case of the algorithm expressed in :

$$\theta(nbProduct \ * \ (nbMargin - 1)^2)$$

So here in $(5*(4-1)^2)$ = 45 steps.

This is quite expensive, especially if we need to discretize the price value in a lot of numbers. A better algorithm, that would learn in each case the best prices configuration and cost less, would be to take the maximum of $X_{j,t} * n_j(X_{j,t})$ for each product independently, resulting in cost of the order of $\theta(nbProduct \ * \ nbMargin)$ but this in each case, even if the best choice is the first price for each product.

Now, we will see how to deal with the greedy algorithm when some of the parameters are not known.

Learning approach: Introduction to the methodology

In order to deal with this issue, we are going to ressort of bandit algorithms techniques. In fact, we have implemented a UCB1 and Thompson Sampling algorithm to learn all the parameters.

They do not differ from the implementation that we have seen in the course but there is just one change. The collected rewards are not stocked in the learner. We have proceeded like this because for each day, we have a different number of realization of what we want to observe (we will be more explicit in the subtitle dedicated to the step), instead of having just one reward in the classical case.

As we have multiple rewards in one day (equivalent to having one reward in more days) we will need less days to learn the parameter. In order to have one reward per day and use it to do an average over multiple experiences, we have done the mean of rewards of that day thus for each day of each experiment, we have one reward.

We do not have stock this in the learners because our environment is designed to return us this reward and instead of stocking it in the learner and then to access them to compute evaluation metrics (regret, reward), we directly compute them when the simulator gives its outputs.

Evaluation

At each step, we compute the regret in order to evaluate the time of our learners to learn the different parameters. As we have said in the methodology, for each day we compute the mean of all the rewards that have appeared in it (but all the rewards are given to the learner so that he can learn efficiently). This allow us to plot two almost equivalent metric that allow us to see the learning process on two aspects :

- the average expected reward with respect to the time : this allows us to observe the expected reward of the arm played at each time and to see how the learner is evolving. We also plot the line of the best arm that allows us to see the convergence at the end and also the regret that is the area between this line and the others rewards. The average is taken over the multiple experiences

- the cumulative average regret : we directly plot the difference between the best expected reward and the one we have for one day with respect to the time. This allows us to better see when the learner has really learned the best arm, thanks to the stagnation of the cumulative regret. We also plot with it the average standard deviation.

All the plots are in the slides