Advertising and Pricing

Scenario. Consider the scenario in which advertisement is used to attract users on an ecommerce website and the users, after the purchase of the first unit of a consumable item, will buy additional units of the same item in future. The goal is to find the best joint bidding and pricing strategy taking into account future purchases.

Environment. Imagine a consumable item (for which we have an infinite number of units) and two binary features. Imagine three classes of customers C1, C2, C3, each corresponding to a subspace of the features' space. Each customers' class is characterized by:

- a stochastic number of daily clicks of new users (*i.e.*, that have never clicked before these ads) as a function depending on the bid;
- a stochastic cost per click as a function of the bid;
- a conversion rate function providing the probability that a user will buy the item given a price;
- a distribution probability over the number of times the user will come back to the ecommerce website to buy that item by 30 days after the first purchase (and simulate such visits in future).

Every price available is associated with a margin obtained by the sale that is known beforehand. Do not need to simulate the functioning of the auctions and the other advertisers.

Steps. You need to complete the following steps.

1. Formulate the objective function when assuming that, once a user makes a purchase with a price *p*, then the ecommerce will propose the same price *p* to future visits of the same user and this user will surely buy the item. The revenue function must take into account the cost per click, while there is no budget constraint. Provide an algorithm to find the best joint bidding/pricing strategy and describe its complexity in the number of values of the bids and prices available (assume here that the values of the parameters are known). In the following Steps, assume that the number of bid values are 10 as well as the number of price values.
2. Consider the online learning version of the above optimization problem when the parameters are not known. Identify the random variables, potential delays in the feedback, and choose a model for each of them when a round corresponds to a single day. Consider a time horizon of one year.
3. Consider the case in which the bid is fixed and learn in online fashion the best pricing strategy when the algorithm does not discriminate among the customers' classes (and therefore the algorithm works with aggregate data). Assume that the number of daily clicks and the daily cost per click are known. Adopt both an upper-confidence bound approach and a Thompson-sampling approach and compare their performance.
4. Do the same as Step 3 when instead a context-generation approach is adopted to identify the classes of customers and adopt a potentially different pricing strategy per class. In doing that, evaluate the performance of the pricing strategies in the different classes only at the optimal solution (*e.g.*, if prices that are not optimal for two customers' classes provide different performance, you do not split the contexts). Let

us remark that no discrimination of the customers' classes is performed at the advertising level. From this Step on, choose one approach between the upper-confidence bound one and the Thompson-sampling one.

5. Consider the case in which the prices are fixed and learn in online fashion the best bidding strategy when the algorithm does not discriminate among the customers' classes. Assume that the conversion probability is known. However, we need to guarantee some form of safety to avoid the play of arms that provide a negative revenue with a given probability. This can be done by estimating the probability distribution over the revenue for every arm and making an arm eligible only when the probability to have a negative revenue is not larger than a threshold (*e.g.*, 20%). Apply this safety constraint after 10 days to avoid that the feasible set of arms is empty, while in the first 10 days choose the arm to pull with uniform probability. Do not discriminate over the customers' classes.
6. Consider the general case in which one needs to learn the joint pricing and bidding strategy under the safety constraint introduced in Step 5. Do not discriminate over the customers' classes both for advertising and pricing.
7. Do the same as Step 6 when instead discriminating over the customers' classes for pricing. In doing that, adopt the context structure already discovered in Step 4.

Duties. You are required to:

- Produce the Python code.
- Produce a technical report describing the environment, the algorithms and the plot representing the regret and the reward of every algorithm. Provide also a practical application motivating the scenario.
- Produce a presentation as a summary of the technical report.

Pricing and Matching

Scenario. Consider the scenario in which a shop has a number of promo codes to incentivize the customers that buy an item to buy a different item. The customers can belong to different classes and the promo codes can provide different discounts.

Environment. Imagine two items (referred to as first and second items; for each item we have an infinite number of units) and four customers' classes. The daily number of customers of each class is described by a potentially different (truncated) Gaussian probability distribution. Each class is also associated with a potentially different conversion rate returning the probability that the user will buy the first item at a given price.

Once a buyer has bought the item, she/he can decide to buy the second item that can be or not promoted. There are four different promos P0, P1, P2, P3, each corresponding to a different level of discount. P0 corresponds to no discount. Given the total number of customers, the business unit of the shop decides the number of promos as a fraction of the total number of the daily customers and is fixed (use two different settings in your experiments that you are free to choose). Each customers' class is also associated with a potentially different conversion rate returning the probability that the user will buy the second item at a given price after she/he has bought the first. The promos will affect the conversion rate as they actually reduce the price.

Every price available is associated with a margin obtained by the sale that is known beforehand. This holds both for the first and the second item.

The conversion rates will change during time according to some phases due to, *e.g.*, seasonality.

Steps. You need to complete the following steps.

1. Provide a mathematical formulation of the problem in the case in which the daily optimization is performed using the average number of customers per class. Provide an algorithm to find the optimal solution in the offline case in which all the parameters are known. Then, during the day when customers arrive, the shop uses a randomized approach to assure that a fraction of the customers of a given class gets a specified promo according to the optimal solution. For instance, at the optimal solution, a specific fraction of the customers of the first class gets P0, another fraction P1, and so on. These fractions will be used as probabilities during the day.
2. Consider the online learning version of the above optimization problem, identify the random variables, and choose a model for them when each round corresponds to a single day. Consider a time horizon of one year.
3. Consider the case in which the assignment of promos is fixed and the price of the second item is fixed and the goal is to learn the optimal price of the first item. Assume that the number of users per class is known as well as the conversion rate associated with the second item. Also assume that the prices are the same for all: the classes (assume the same in the following) and that the conversion rates do not change unless specified differently below. Adopt both an upper-confidence bound approach and a Thompson-sampling approach and compare their performance.
4. Do the same as Step 3 when instead the conversion rate associated with the second item is not known. Also assume that the number of customers per class is not known.
5. Consider the case in which prices are fixed, but the assignment of promos to users need to be optimized by using an assignment algorithm. All the parameters need to be learnt.
6. Consider the general case in which the shop needs to optimize the prices and the assignment of promos to the customers in the case all the parameters need to be learnt.
7. Do the same as Step 6 when the conversion rates are not stationary. Adopt a sliding-window approach.
8. Do the same as Step 6 when the conversion rates are not stationary. Adopt a change-detection test approach.

Duties. You are required to:

- Produce the Python code.
- Produce a technical report describing the environment, the algorithms and the plot representing the regret and the reward of every algorithm. Provide also a practical application motivating the scenario.
- Produce a presentation as a summary of the technical report.

Advertising and Social influence

Scenario. Consider the scenario in which an advertiser tries to activate a cascade in a social network by using sponsored ads. On the other side, the social network platform works as an ad publisher. The nodes of social networks can be of different categories and the activation probabilities depend on the categories of the nodes.

Environment. Imagine a social network in which the nodes belong to 5 categories and in which the activation probabilities associated with the edges are linear functions in the features. Every category is used as a different advertising target, corresponding to 5 different campaigns. Each campaign is a part of a different auction. More precisely, for every category, we have an auction returning an ad allocation and this allocation is shown to every advertiser of that category. The click behavior of each node is independent from the others. Model these auctions such that there is a single (learning) advertiser, while the other advertisers have a stochastic behavior. Adopt a classical VCG approach, with 6 slots and where there are no budget constraints. In every auction, there is a finite number of bids (4 strictly positive values and 0, corresponding to turn off the campaign).

At every round, the advertiser can bid for their ads for every category of users, the platform displays an ad allocation to every node accordingly, the users can click or not the ad, and, in the case of click, a cascade induced by social influence is activated. The feedback is edge level. Remarkably, a node behaves as a social influencer only if it clicked the ad, and every node can be (with a given probability) a social influencer thus leading to situations in which multiple nodes generate influence. Although we have 5 different auctions, the message is the same and therefore the cascade will be the same. (A suggestion to deal with social influencers that activate with a given probability: for every node, add a fictitious node, connect it to the "true" node, and its activation probability will be equal to the click probability).

Assume that the stochastic behavior of the advertisers is non-stationary. In particular, assume that the changes are abrupt in some phases.

Steps. You need to complete the following steps.

1. Estimate the social influence generated by a set of bids with Monte Carlo methods and show how the approximation error varies as the number of repetitions varies. More precisely, evaluate the average number of activated nodes as the number of iterations varies. Use the best performance you obtain as a baseline and show the performance gap due a different number of iterations (also report the theoretical upper bound on the approximation error). Notice, for every node we have an auction (therefore the number of auctions is equal to the number of nodes), but a bid applies to all the nodes of the same category (therefore at every round the advertisers makes 5 bids). The mapping between nodes and categories is known.
2. Formulate the optimization problem in which the objective function is the difference between the expected number of activated nodes (multiplied by some opportune coefficient) and the expected total cost per click (summed over all the nodes). Define which are the random variables and choose how to model them.

3. Design the following greedy optimization algorithm. For every category, sort the bids from 0 to the largest and consider them as increasing bidding levels. Assume to have a vector of 5 bids, one per category. Evaluate the marginal gain one can obtain by increasing the bid by one level for every category (that is, increase a single bid by a single level and evaluate the marginal gain, do that for every category, and finally choose the increase maximizing the marginal gain and apply it; then reiterate the procedure while there is at least one positive increase).
4. Address the online learning of the best bids when the slots' observation probabilities are known, while the click probabilities and the cost per click are unknown. Adopt a classical bandit algorithm in which one can learn the click probability (quality) of an ad only when the ad is displayed in the first position and compare its performance with a cascading bandit algorithm in which the click probability can be learnt by using any samples, including when the ad is not displayed in the first position. Assume the activation probabilities to be known and the behavior of the stochastic advertisers to be stationary. Adopt both an upper-confidence bound approach and a Thompson-sampling approach and compare their performance.
5. Do the same as Step 4 when the activation probabilities are not known.
6. Do the same as Step 5 when the behavior of the advertisers is non-stationary by using sliding-window approaches.

Duties. You are required to:
- Produce the Python code.
- Produce a technical report describing the environment, the algorithms and the plot representing the regret and the reward of every algorithm. Provide also a practical application motivating the scenario.
- Produce a presentation as a summary of the technical report.