Project: **Pricing & Advertising**

Team members: **Mattia Bosio, Giacomo Lodigiani, Matteo Orsolini**

Politecnico di Milano

Academic Year 2019/2020

POLITECNICO
MILANO 1863

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Problem description

Problem

We have to model a scenario in which a seller exploits advertising tools to attract more and more users to its website, thus increasing the number of possible buyers. The seller needs to learn simultaneously the conversion rate and the number of users the advertising tools can attract.

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Our setting

Setting description

We imagined a website selling streaming subscriptions to basketball events, and identified three possible classes of target users, and each one can have a targeted campaign:

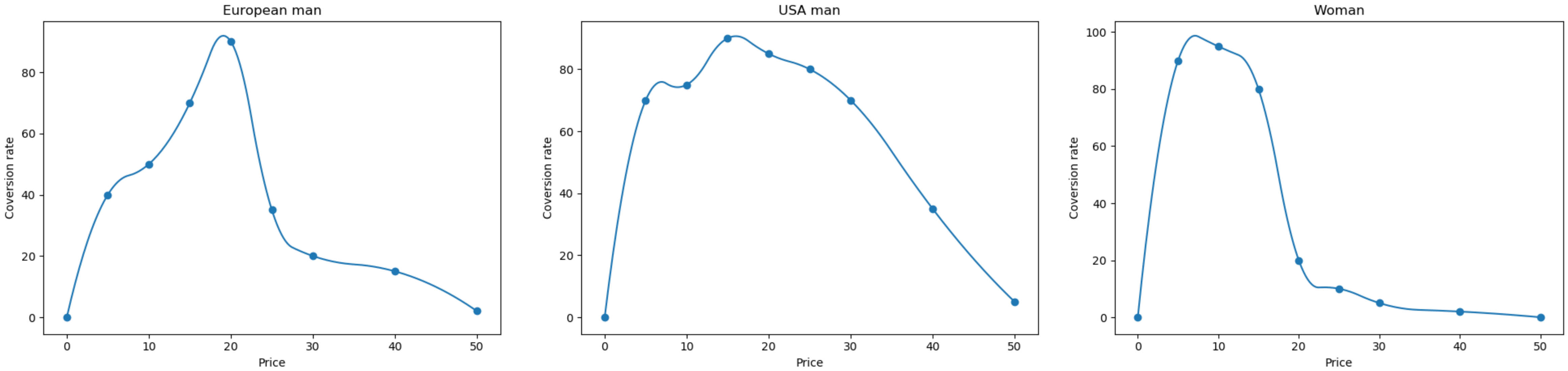
- European men
- USA men
- Women

POLITECNICO
MILANO 1863

Our setting

Convertion rates

Of course the different classes of users, having different interests, will have different conversion rates.



Conversion rates for the 3 classes of users

Our setting

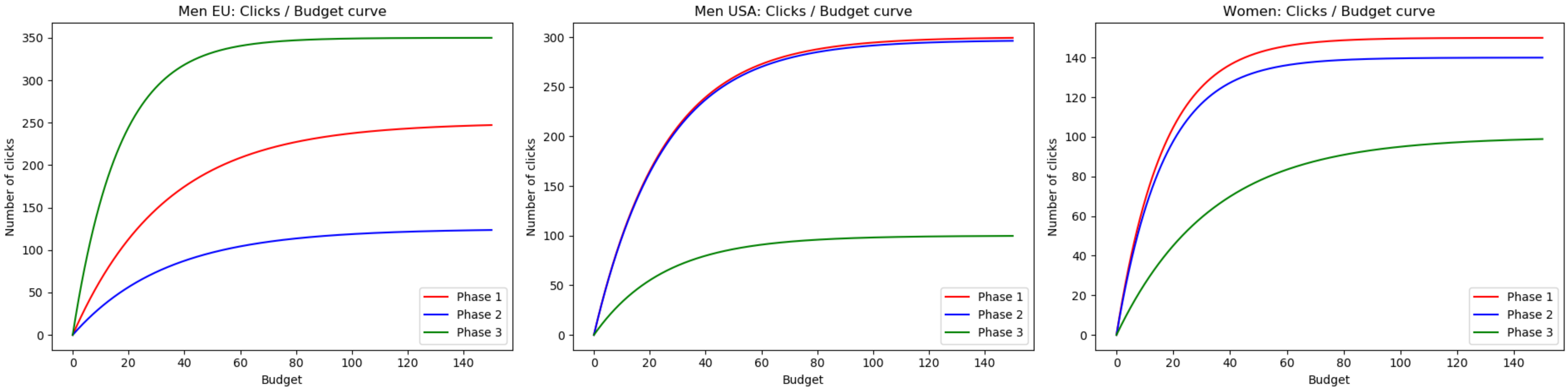
Abrupt phases

We also imagined different abrupt phases changing the market and subsequently the number of users clicking on ads for a given budget:

- Phase 1: the launch of the product
- Phase 2: the enter in the market of a new competitor
- Phase 3: the updating of the product with new features

POLITECNICO
MILANO 1863

Our setting



Clicks / Budget curves for the 3 user classes and the 3 phases

POLITECNICO
MILANO 1863

Index

POLITECNICO MILANO 1863

Single-phase budget allocation

Our Approach: "Gaussian Process Thompson Sampling".

- Perform regression to estimate the number of clicks per day with respect to the budget allocation of each of the three sub-campaigns.
- Use Gaussian Processes in order to obtain a probability distribution over the target value, the number of clicks.
- Apply Thompson Sampling algorithm pulling the arms (budget allocation) from the estimated probability distribution.
- Solve the combinatorial part of the mab problem using the knapsack tabular algorithm.
- Get the realization of the allocation selected, thanks to the Knapsack algorithm, and update the Gaussian Processes.

POLITECNICO
MILANO 1863

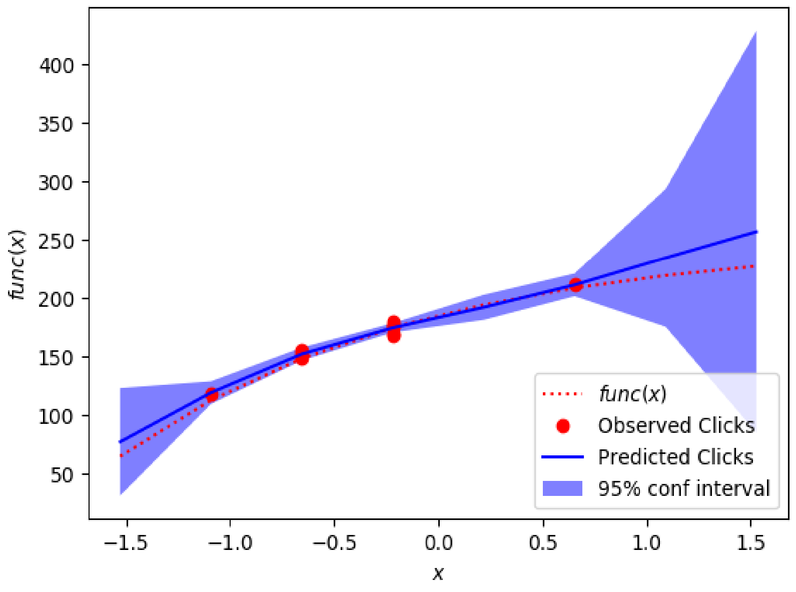
Single-phase budget allocation

Some Difficulties: Kernel Parameters

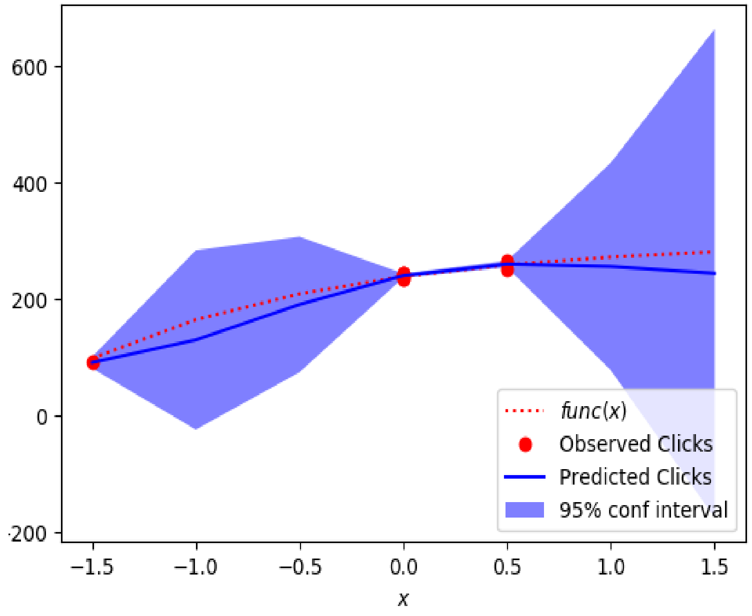
- We use the product of the Constant Kernel and RBF kernel (defined in "sklearn" library).
- ***constant_value * exp(- d(x_i, x_j)^2 / 2l^2)***, where d(,) is the Euclidean distance.
- To estimate them we fit 1000 samples for each function using 'L-BGFS-B' optimizer (log marginal likelihood)
- We keep them fixed in our experiment in order to achieve better result from the start and reduce the computational time.

POLITECNICO
MILANO 1863

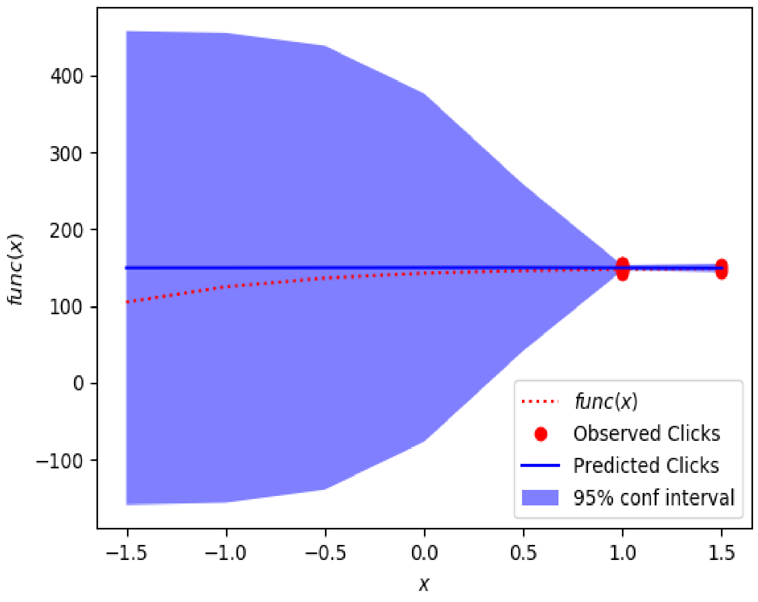
Single-phase budget allocation



Men Eu



Men USA



Women

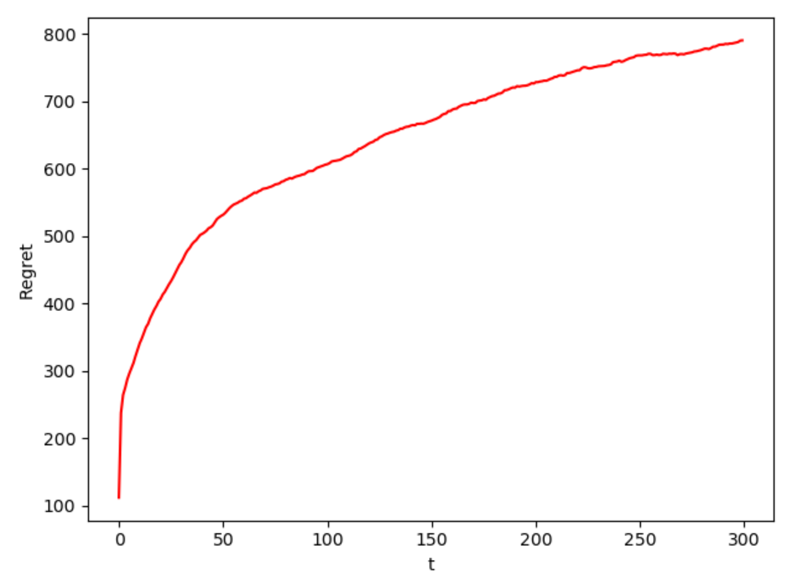
The Gaussian Processes fitting the corresponding functions after 10 days

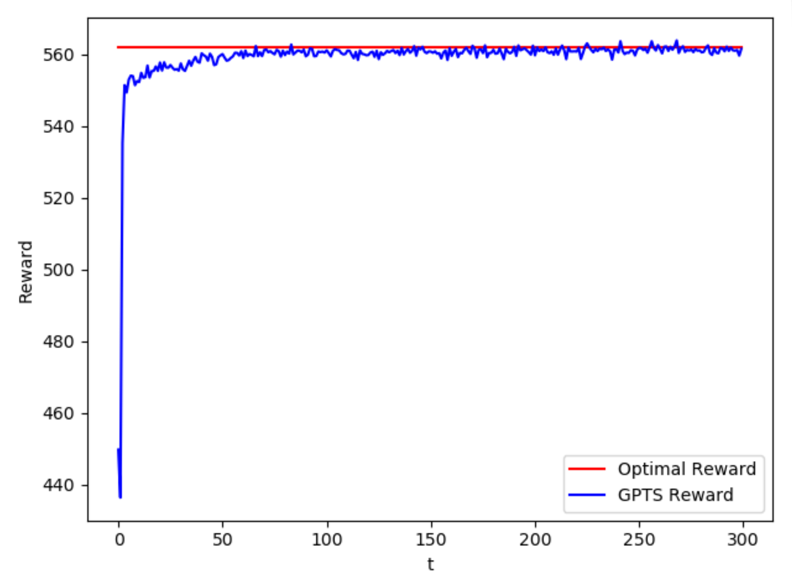
POLITECNICO
MILANO 1863

Single-phase budget allocation

Results

Here we can see the regret, which is less than linear as expected, and the reward, which converges to the optimum





Regret and Reward obtained in advertising with 100 experiments and 300 days as T horizon

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Multi-phase budget allocation

Our Approach: sliding-window combinatorial bandit algorithm

- We adjust the current algorithm using a sliding-window technique.
- Fixed the window size, the Gaussian Processes use only the last n samples inside the sliding window, to fit the function and provide new estimates.
- In this way, being in an non-stationary environment, our learners can “forget” about the first samples that are no more relevant and base the next estimates on new samples that are representative of the changed environment.
- The kernel parameters are estimated using the same approach for each of the 3 phases. The model uses the same fixed parameters for all the phases: we have used the mean value on the phases.

POLITECNICO
MILANO 1863

Multi-phase budget allocation

Different Result

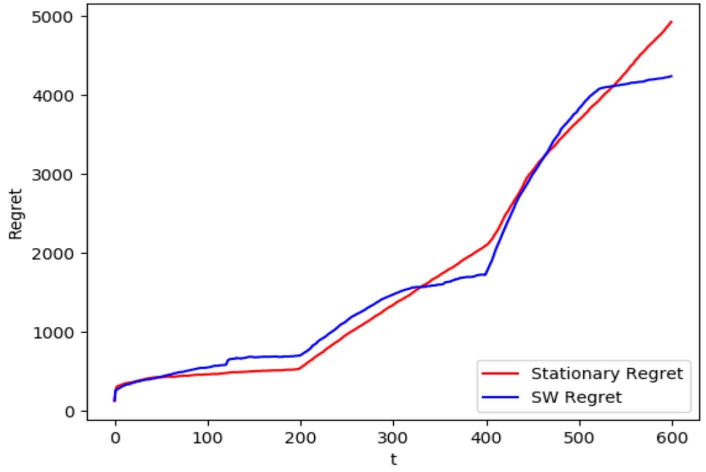
- The size of the sliding window, with respect to the length of the abrupt phase, appears to be crucial in the effectiveness of the algorithm.
- It must be enough big in order to be able to guarantee a good number of samples, which means a good estimate of the real function, but at the same time it can not be too long, since it has to capture the changes of the environments.
- We set it in the order of $\Theta(\sqrt{T})$, more precisely constant * $\sqrt{T}$, trying different constants.

POLITECNICO
MILANO 1863

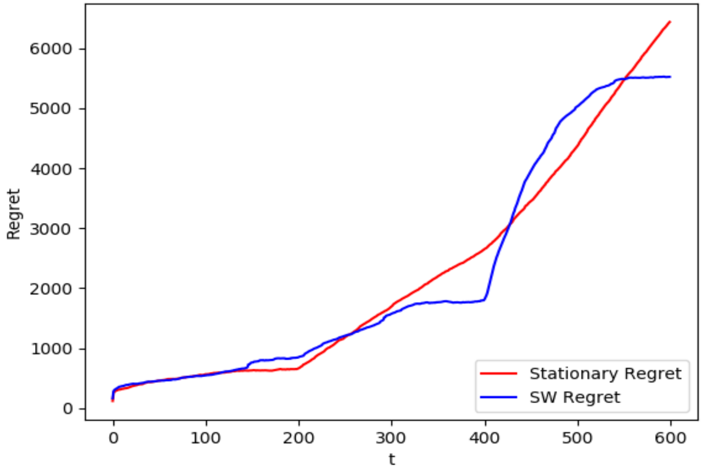
Multi-phase budget allocation

Different size of the sliding-window

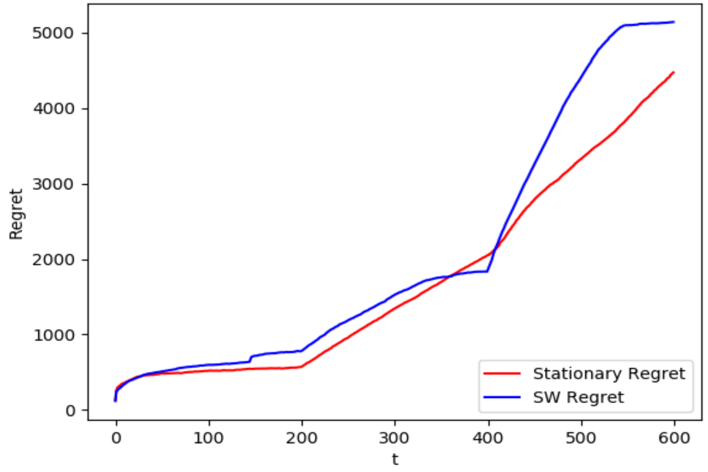
This is the regret corresponding to the different window-size. Each plot is the result of 150 experiments. The three phases start at time step 0, 200 and 400 with a corresponding length of 200 days.



$4 * \sqrt{T}$



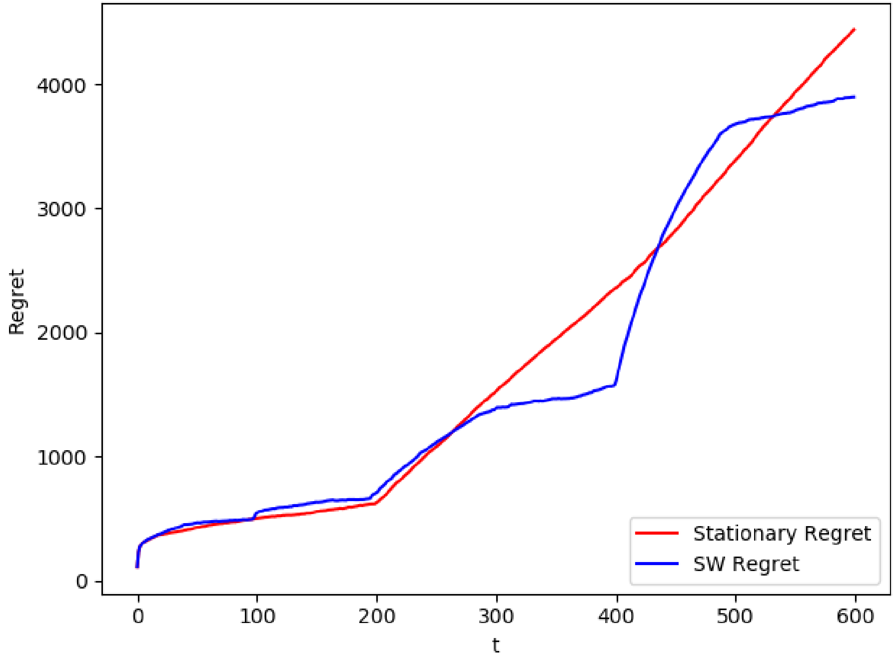
$5 * \sqrt{T}$



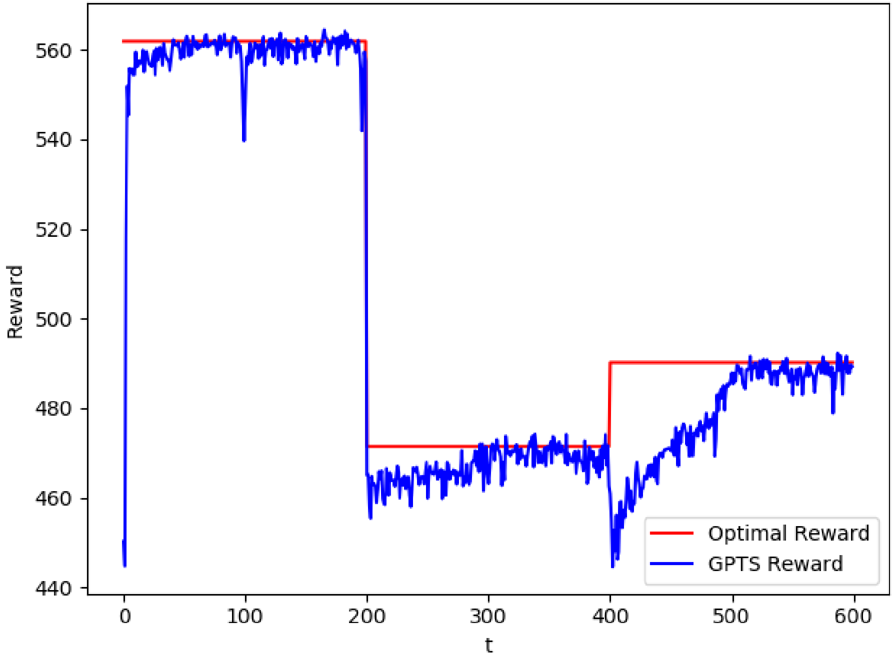
$6 * \sqrt{T}$

POLITECNICO
MILANO 1863

Multi-phase budget allocation



4 * √T wins!



Regret and Reward obtained in sliding window advertising with 100 experiments and a window size of 200.

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Learning algorithm for pricing

Found the optimal price

- The environment is defined by the different conversion rate curves, one for each type of user.
- The number of users is defined by the daily number of clicks, generated by the advertising environment with a fixed budget allocation.
- In this scenario the seller can not perform pricing discrimination therefore has to sell to the same price to all the users.
- The clairvoyant is defined as the best price in the aggregate conversion rate curve.

POLITECNICO
MILANO 1863

Learning algorithm for pricing

Thompson Sampling Algorithm

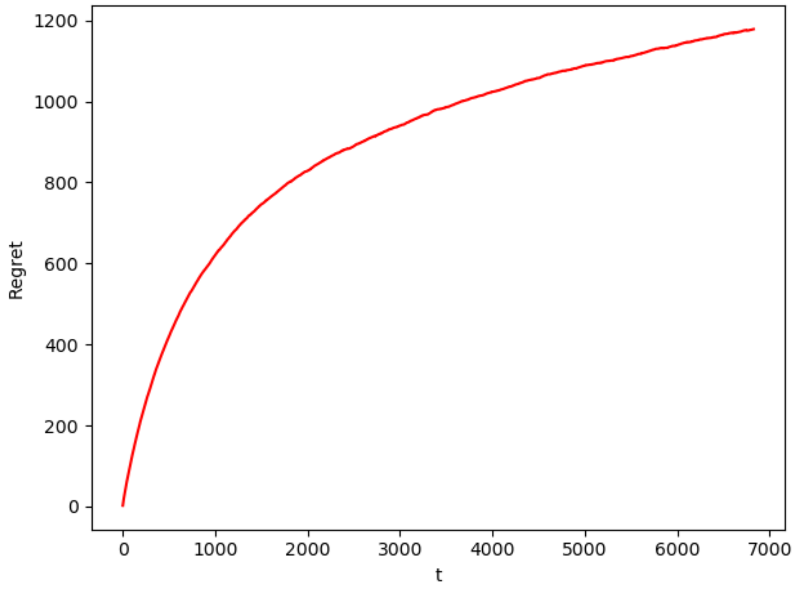
- To solve this Multi Armed Bandit Problem, and found the optimal price, we use Thomspon Sampling Algorithm.
- The random variable is the conversion rate of each arm, so what we can observe is a Bernulli reward, for this reason we use a Beta distribution as prior.
- The expected reward of an arm is represented by the product between the conversion rate and the price itself.
- For each user we choose the price sampling the beta distribution and getting the one with highest reward. We observe the realization and then we update the distribution.

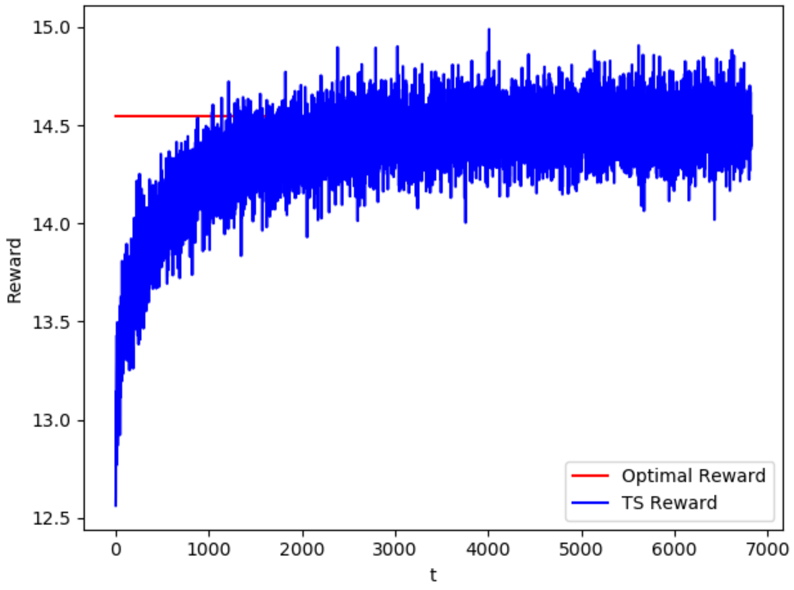
POLITECNICO
MILANO 1863

Learning algorithm for pricing

Results

In the few days, since we interact with a huge number of users in a day, we are able to get the optimal price.





Regret and Reward obtained in pricing with 10000 experiments and 14 days as T horizon

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Context generation algorithm

Our approach

For each context inside the partitions, generated by the three sub campaigns we have considered in the advertising problem, we have created a learner which, by applying the Thompson Sampling algorithm, estimates the expected reward, that is the profit, associated to each price for that specific context.

The setting we have used is the following:

- The daily number of users is generated according to the advertising problem and the proportion of the users is respected too.
- The experiments last 10 weeks.

POLITECNICO
MILANO 1863

Context generation algorithm

Considerations (first algorithm)

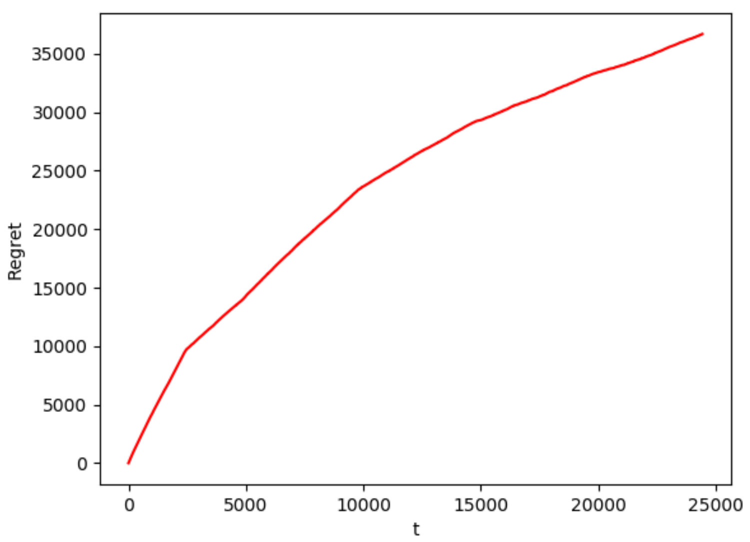
- Since in expectation the daily number of users, divided in proportion to the probability of their context, is slightly lower than 500, we have observed that after one week, thanks to the huge number of samples observed, performing price discrimination on the three targets of the sub campaigns is , with good probability, identified as the best possible choice since we usually reach a very good approximation of the best expected reward for each target.
- The problem is that observing this huge amount of sample ruins the performance of the algorithm in later weeks: the bounds are ruined and so the algorithm tends to choose the action of re-aggregation as a plausible option. Our solution to this issue was to use a stopping criterion: once the algorithm chooses to disaggregate, it can only choose as active partition, a partition with a cardinality strictly larger than the currently active one.

POLITECNICO
MILANO 1863

Context generation algorithm

Results

Here we can see the regret of the first approach.



Regret of first context generation algorith with 150 experiments

POLITECNICO
MILANO 1863

Context generation algorithm

Our approach (second algorithm)

Thanks to the fact that the daily number of samples is consistent, we have implemented a " sequential" version of the algorithm using Ɛ-greedy exploration:

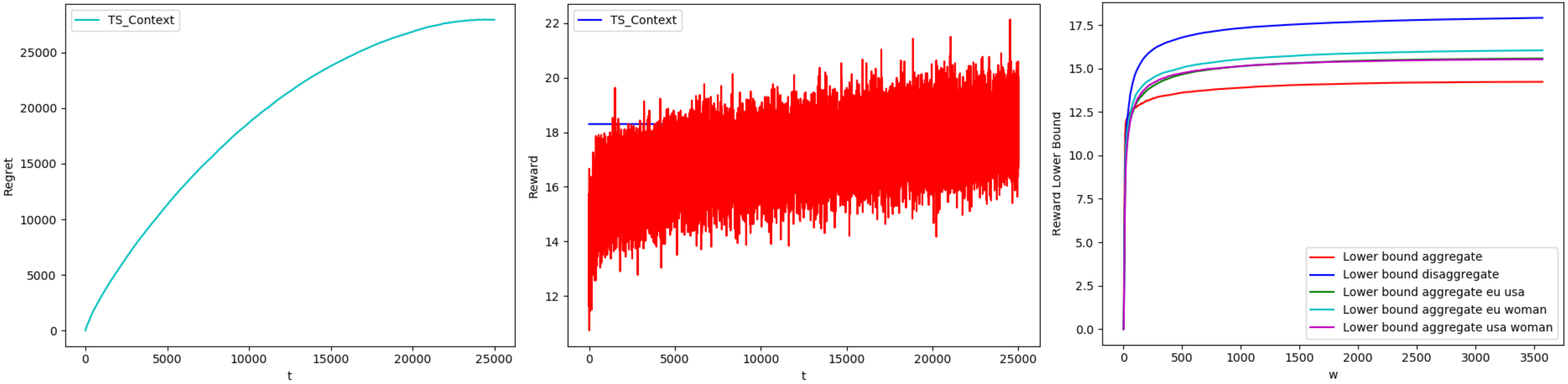
- Every week we only get samples relative to the active context (represented by an aggregate learner)
- At the end of every week we choose the active context for the next one in the following way: random with probability Ɛ, argmax(lower bound) with probability 1 – Ɛ
- Ɛ decreseas as we approach the end of the time horizon T (Ɛ = 1 – t/T)

POLITECNICO
MILANO 1863

Context generation algorithm

Results

Here we can see the regret, reward and bounds of the second approach.



Regret, Reward and Bounds obtained with the second context generation algorithm with 150 experiments

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Budget and pricing optimization (multiple prices)

Our approach

Since each subcampaign targets a single class of users, the two problems, pricing and advertising, can be decomposed. So at every iteration, first we run our pricing algorithm (with the percentage of users belonging to each class depending on the budget allocation) to find the values per click (conversion rates), and then we run our advertising algorithm to improve budget allocation.

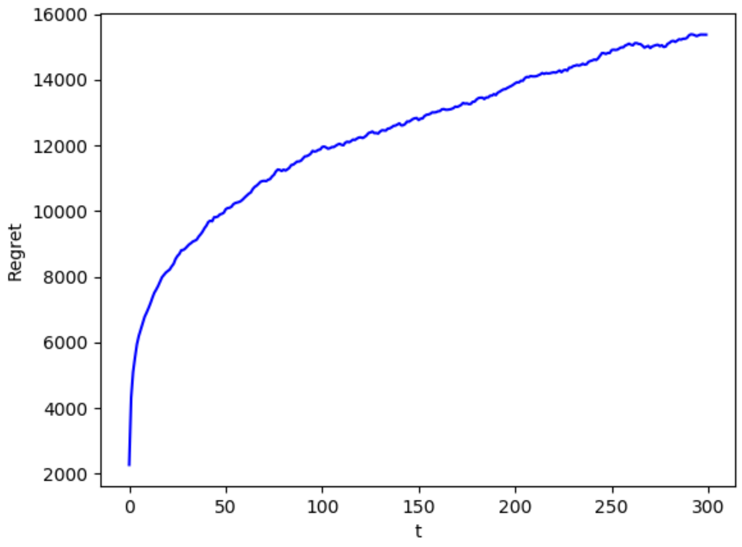
Additional comments

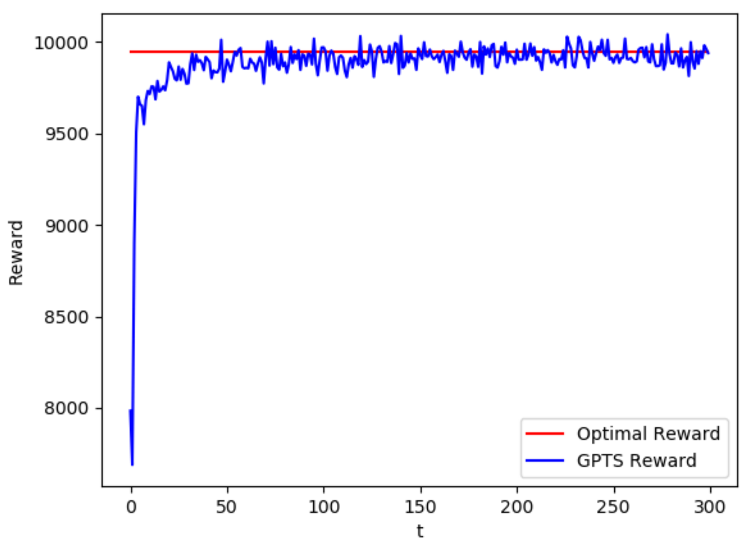
We developed two versions of the algorithm: in the first one we used for the next iteration the prices corresponding to the best budget allocation (the one that maximises number of clicks * value per click) found in previous one; in the second version we just used the best budged allocation found in previous iteration to estimate incoming clicks in the next one, but here the new prices don't have to be the ones corresponding to previous best budget allocation.

Budget and pricing optimization (multiple prices)

Results

Here we can see the regret and the reward of the first implementation, where the price for each context corresponds to the price which conversion rate guarantees best budget allocation in previous iteration.



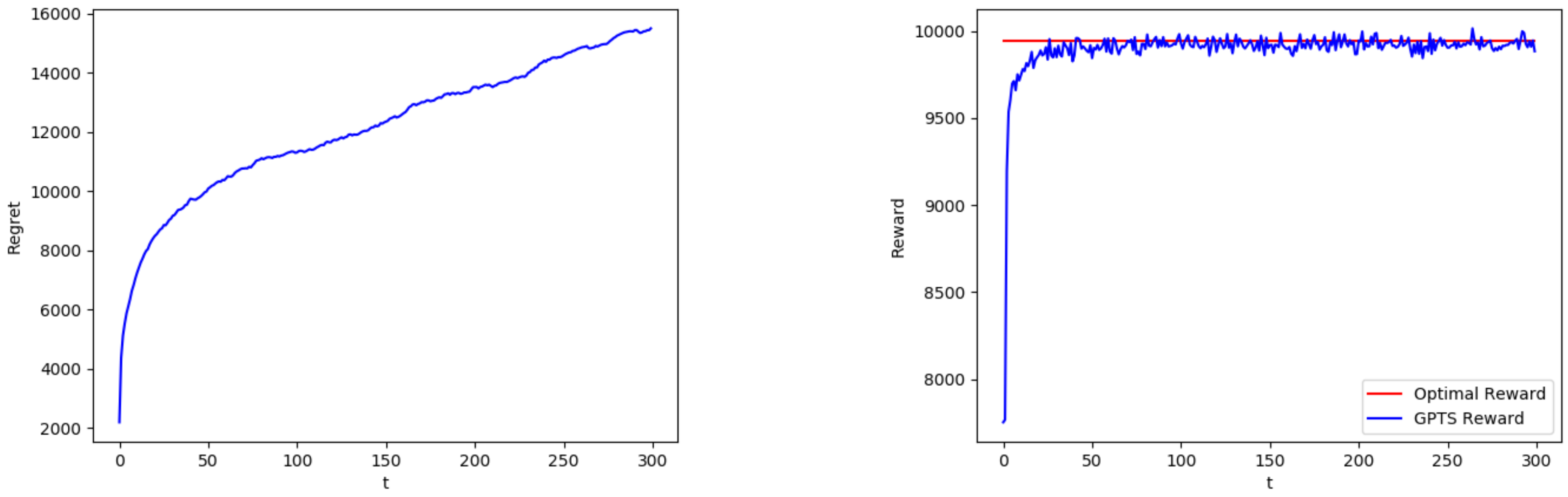


Regret and Reward obtained with fixed price and 100 experiments

Budget and pricing optimization (multiple prices)

Results

Here we can see the regret and the reward of the second implementation. The price for each context is chosen during the day according to Thompson sampling learner.



Regret and Reward obtained with non fixed price and 100 experiments

POLITECNICO
MILANO 1863

Index

POLITECNICO
MILANO 1863

Budget and pricing optimization (single price)

Our approach

We used the same iterative algorithm seen in previous point, but here, instead of finding the best price for each class of users, we tried all the possible unique prices to charge to all the users without discriminating. We use an aggregate (unique) learner to interact with the environment, and 3 different learnes to estimate the conversion rates (values per click) for each class of users.

Additional comments

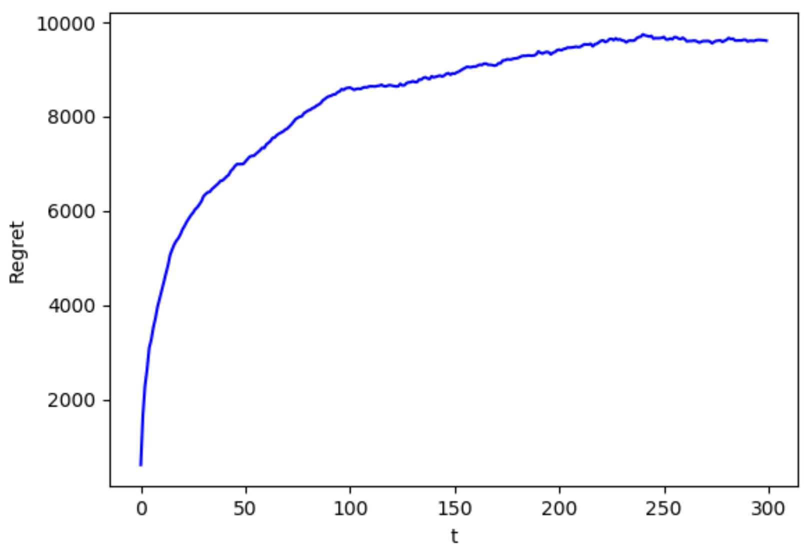
Having one price for all the users, to find the best budget allocation, we run the knapsack tabular algorithm for each price (each price corresponds to different values per click) to find the budget maximizing number of clicks * value per click. Also here we compared two version of the algorithm, one in which the price corresponding to best budget allocation obtained in previous iteration is used in the next one, and one removing this constraint.

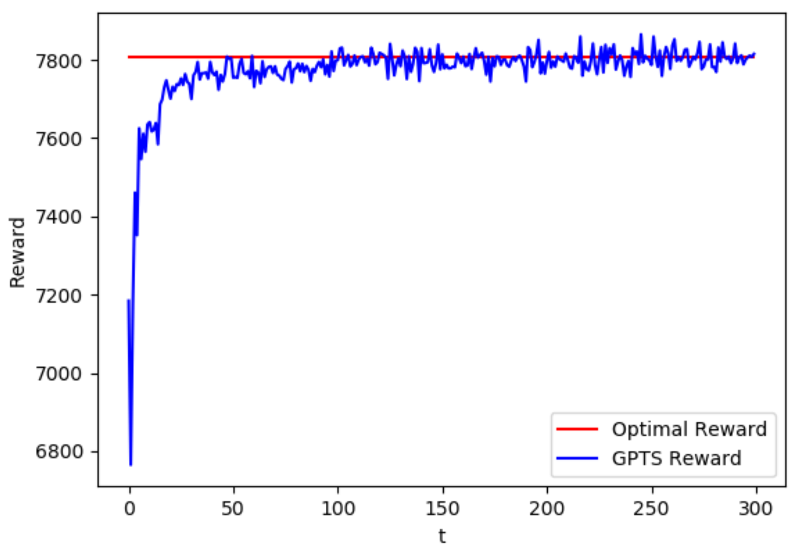
POLITECNICO
MILANO 1863

Budget and pricing optimization (single price)

Results

Here we can see the regret and the reward of the first implementation.





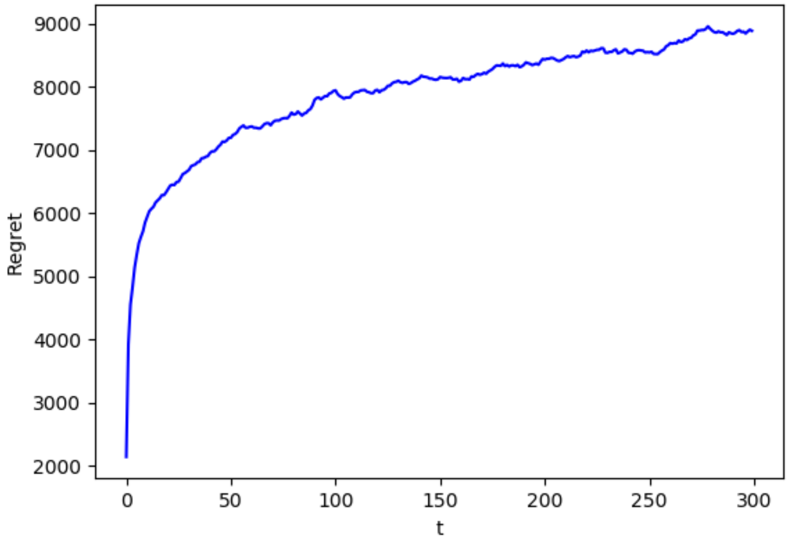
Regret and Reward obtained with fixed price and 75 experiments

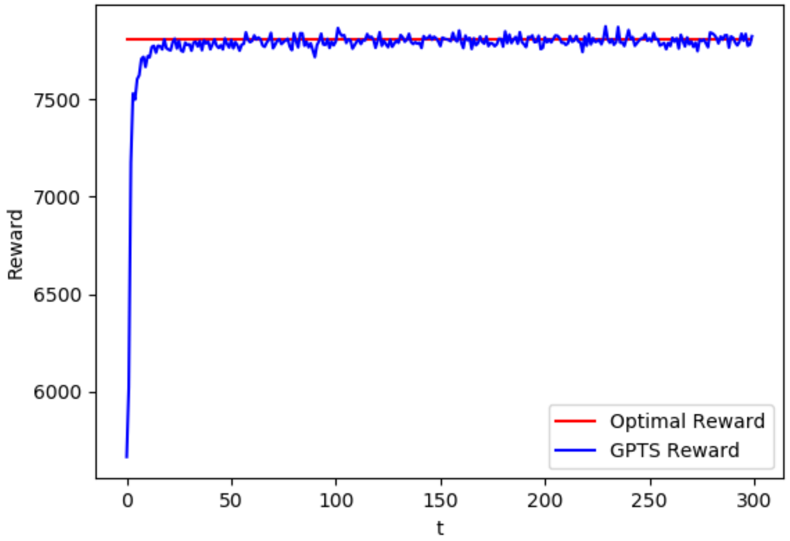
POLITECNICO
MILANO 1863

Budget and pricing optimization (single price)

Results

Here we can see the regret and the reward of the second implementation.





Regret and Reward obtained with non fixed price and 75 experiments

POLITECNICO
MILANO 1863

Thanks for your attention!

POLITECNICO
MILANO 1863