POLITECNICO
MILANO 1863

eit Digital

NAIk

JUST LEARN IT.

Online Learning Applications Project

Pricing and Advertising

A.Y. 2022/2023

Valeria Amato - 10641790
Beatrice Insalata - 10708628
Emanuele Paci - 10681377
Matteo Pancini - 10656944
Andrea Riboni - 10699906

MEET OUR AMAZING TEAM

Five EIT Digital Data Science students

Valeria Amato

Beatrice Insalata

Emanuele Paci

Matteo Pancini

Andrea Riboni

POLITECNICO MILANO 1863

PROJECT OVERVIEW

The choice of the product was made considering its relevance in online sales, the potential for seasonality, and the possibility to appeal to different kinds of users. Mountain trekking shoes are a valuable product that stays on a relatively stable price range.

Medium-quality, fashionable design shoes were considered to be sold in the interval:

[50,100,150,200,250]

Users' reserve price can be related to different factors

- **SEASONALITY**
- **ONLINE MARKET**
- **PERSONAL INTEREST**

USER CLASSES

The user classes were defined considering relevant characteristics that can influence the level of attractiveness of the product.

They were divided in three main categories considering two main characteristics:

AGE

Younger users tend to have a higher level of activity

PERSONAL INTEREST

Users who have an interest in outdoor activities

YOUNG & SPORTY

The optimal target for our product.

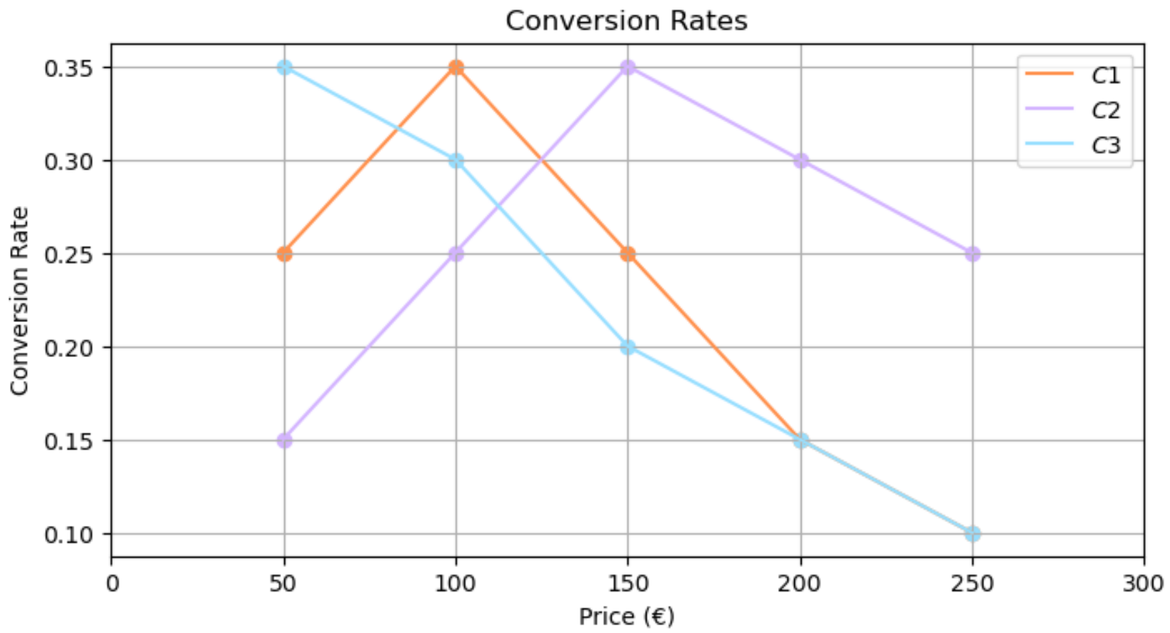
ADULT & SPORTY

A valuable customer to address.

YOUNG AND NON-SPORTY

A great potential public for our offer.

CONVERSION RATES



A PRICE-QUALITY TRADE-OFF

Sporty users are skeptical and will less likely buy a product if its price is too low.

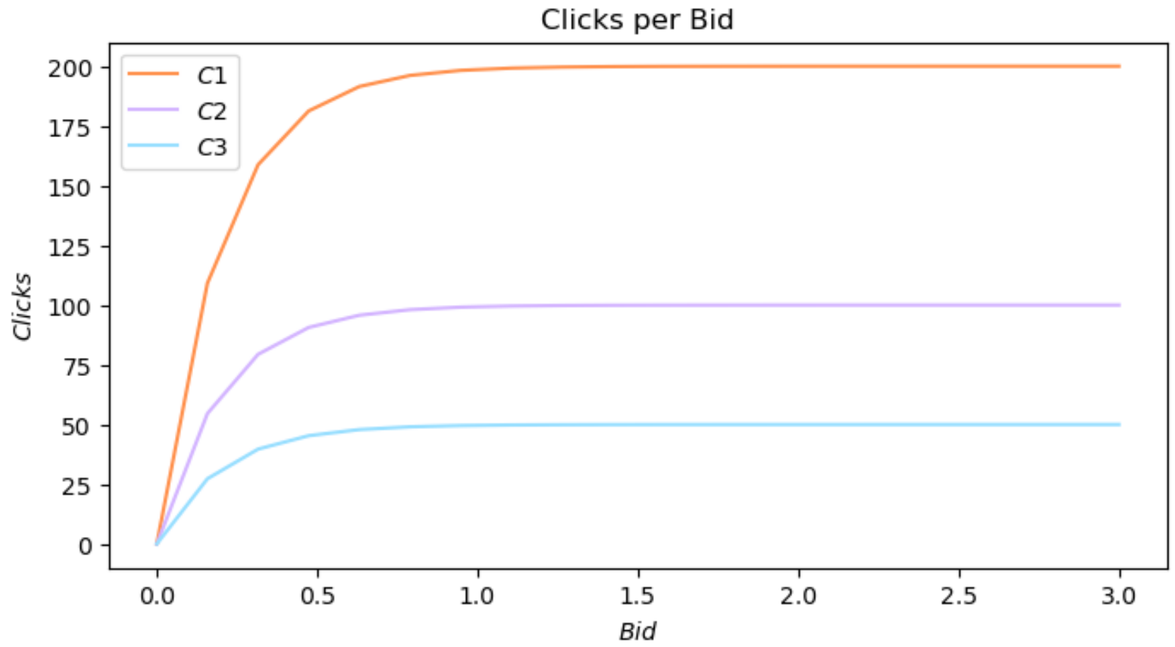
Non-sporty users are less interested in the quality of the product and are more willing to buy the shoes even when the price is *too low*.

YOUNG & SPORTY

ADULT & SPORTY

YOUNG AND NON-SPORTY

AMOUNT OF CLICKS



AN EXPONENTIAL APPROACH

The three classes are represented by similar behaviors for this curve, where sporty-users are more likely willing to click an ad. Being tech-savvy, youngsters are the ones with the highest clicks-per-bid ratio.

$clicks(bid)=200(1-e^{-5[bid]})$

YOUNG & SPORTY

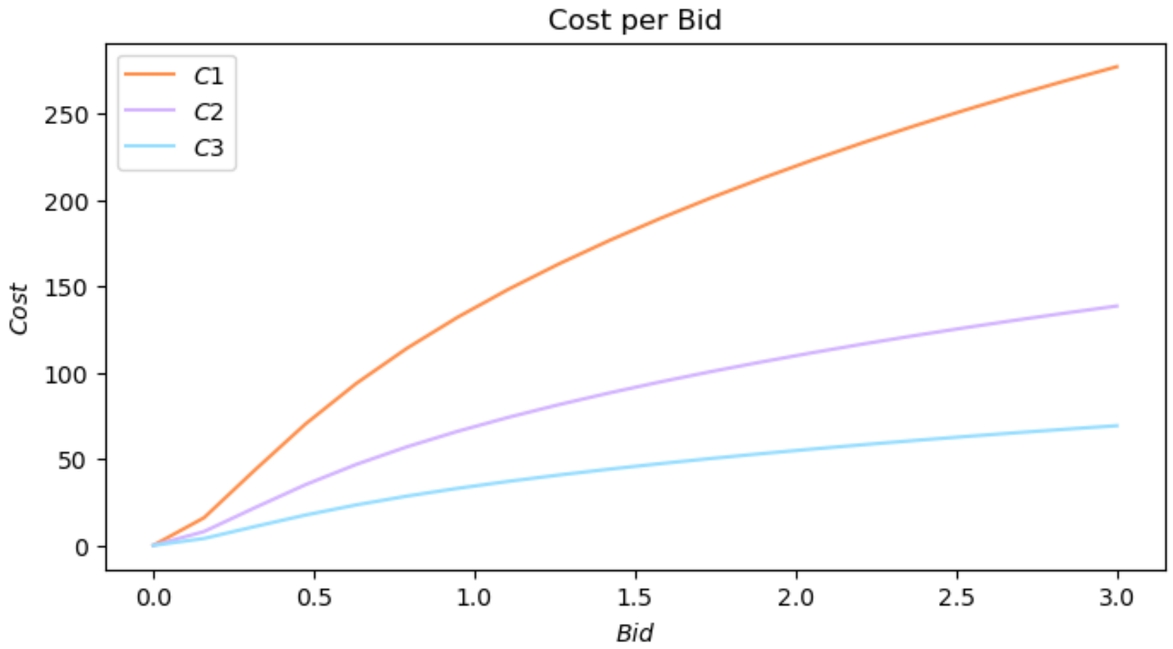
$clicks(bid)=100(1-e^{-5[bid]})$

ADULT & SPORTY

$clicks(bid)=25(1-e^{-5[bid]})$

YOUNG AND NON-SPORTY

COST PER BID



A LOGARITHMIC APPROACH

The cost per bid is expressed by a product.

The total cost for a bid *b* is given by

$$cost(b)=log(b+1)*clicks(b)$$

Therefore the cost curve is based on the click curve, but scales also logarithmically with the bid value.

YOUNG & SPORTY

ADULT & SPORTY

YOUNG AND NON-SPORTY

THE CLAIRVOYANT

COMPUTING THE OPTIMAL PARAMETERS

To determine the optimal bid and the optimal price, we resort to the exact method, which is the exhaustive search over all possible combinations.
This is repeated for each class of users.

3	5	100
User Classes	**Possible Prices**	**Possible Bids**
	[50, 250]	(0, 3]
	50-step	100-step

We decided that the manufacturing cost for the NAIK shoes is 30% of the selling price
Therefore, the margin is constant: 70% of the selling price

POLITECNICO MILANO 1863

STEP 1: LEARNING FOR PRICING

SCENARIO

ALL USERS BELONG TO CLASS C1

PRICING CURVES ARE UNKNOWN

ADVERTISING CURVES ARE KNOWN

REQUEST

TS

APPLY TS ALGORITHM

UCB

APPLY UCB1 ALGORITHM

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

Introducing: UCB1 and TS

- For this project, we are going to use algorithms designed to work in MAB settings
- Mainly, UCB1 and TS

- We use both algorithms to estimate the conversion rates
- They present some key differences

UCB1

- Deterministic
- Pulls arms based on their upper confidence bound

TS

- Probabilistic, Bayesian
- Pulls arms based on a posterior probability

STEP 1: SETUP

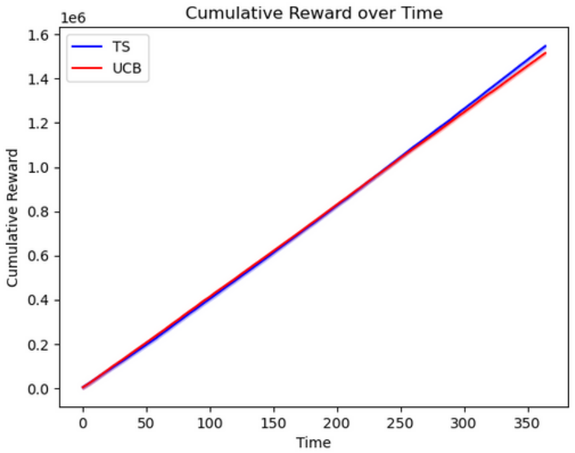
- The advertising environment is known.
- We consider the bid value that maximizes the reward, according to the clairvoyant algorithm

- To average the results, 1000 experiments are ran over the 365 days.
- Two learners are deployed: UCB1 and TS
- In each round, the learners choose the bid that maximizes the product between the estimated conversion rate and the margin, for this class of users.

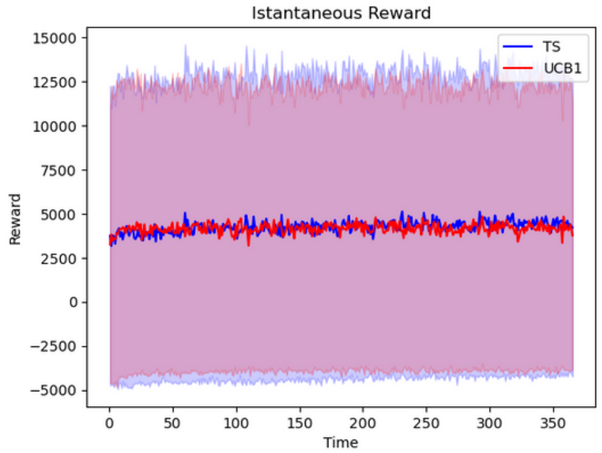
- The advertising environment provides methods for accessing the functions related to the bid value (generating observation rounds)
- The pricing environment gives access to conversion rates and rewards per round
- The TS Learner contains the logic of the algorithm, pulling arms and updating the distribution parameters.

POLITECNICO MILANO 1863

STEP 1: RESULTS

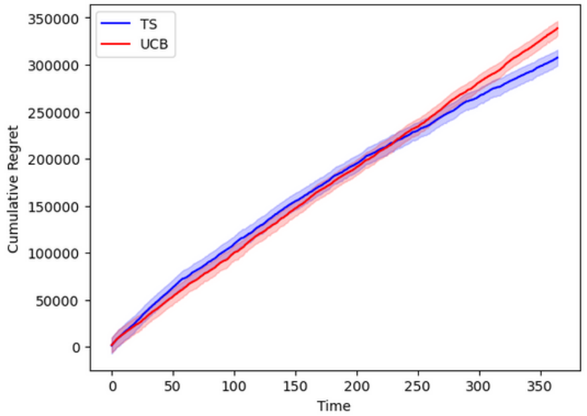


Cumulative reward

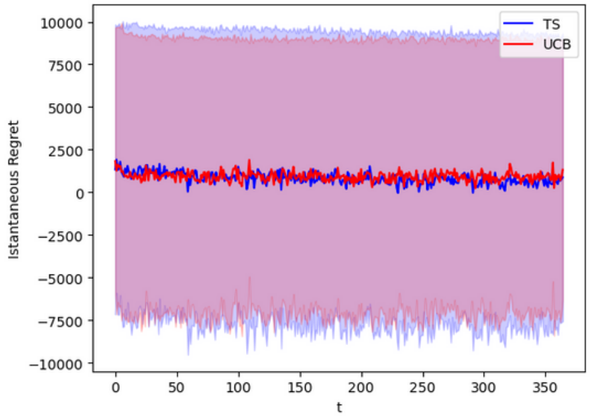


Istantaneous reward

- Both TS and UCB1 achieve a sublinear regret
- TS has a slightly higher reward than UCB1's, which is also more stable



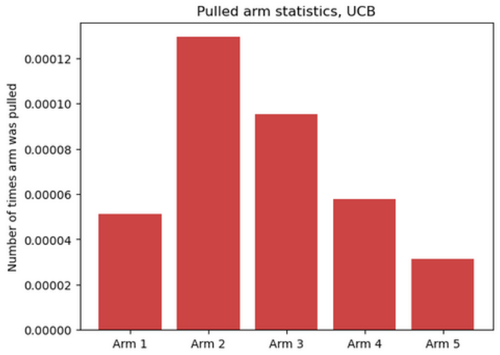
Cumulative regret

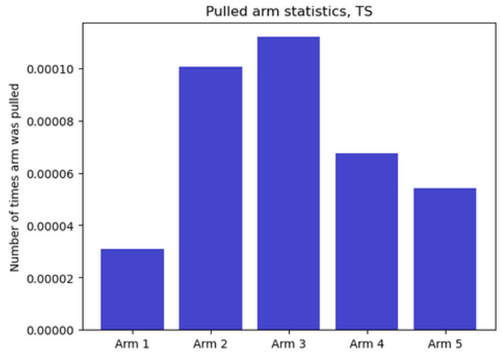


Istantaneous regret

- Given that, TS also has a lower cumulative regret, performing better, in the long run, thanks to the exploration-exploitation tradeoff.

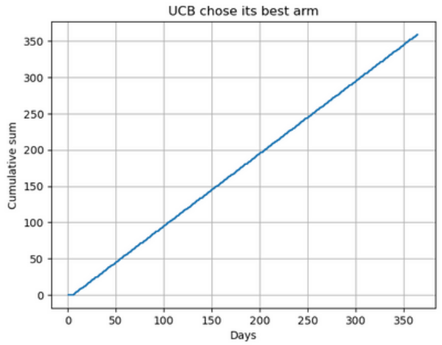
STEP 1: IN DEPTH ANALYSIS

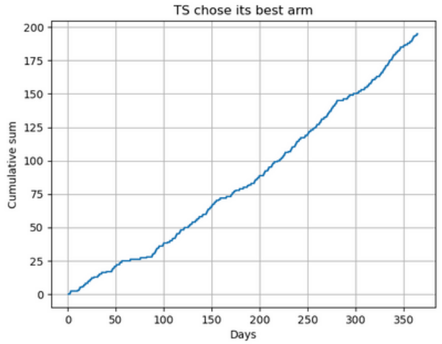




If we consider the number of times each arm has been pulled, we see that the two learners have converged towards different arms.

- UCB1 towards arm 2
- TS towards arm 3



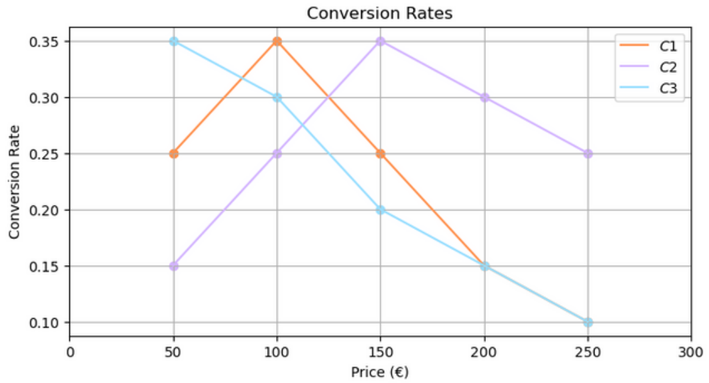


By plotting all the times UCB1 has pulled arm 2 and TS has pulled arm 3 over the time horizon, we can truely observe that TS explores the other arms much more

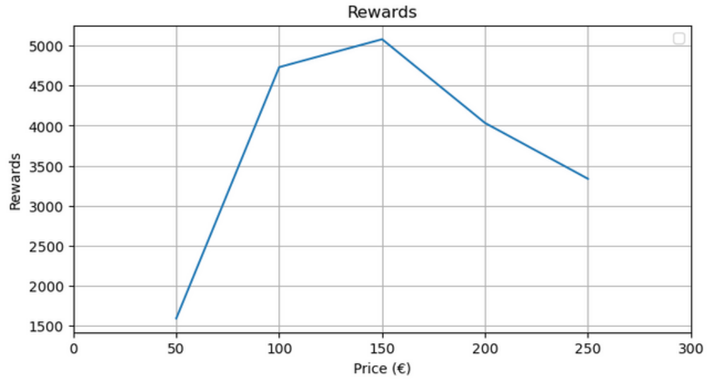
STEP 1: IN DEPTH ANALYSIS

As we have seen, TS achieves slightly better result in this scenario.

This makes sense: the clairvoyant algorithm identified arm 3 to be the best. In fact,



Even though arm 2 has the greatest class C1 conversion rate,



Arm 3 has the highest reward.

POLITECNICO MILANO 1863

STEP 2: LEARNING FOR ADVERTISING

SCENARIO

ALL USERS BELONG TO CLASS C1

PRICING CURVES ARE KNOWN

ADVERTISING CURVES ARE UNKNOWN

REQUEST

TS

APPLY GP-TS ALGORITHM

UCB

APPLY GP-UCB ALGORITHM

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

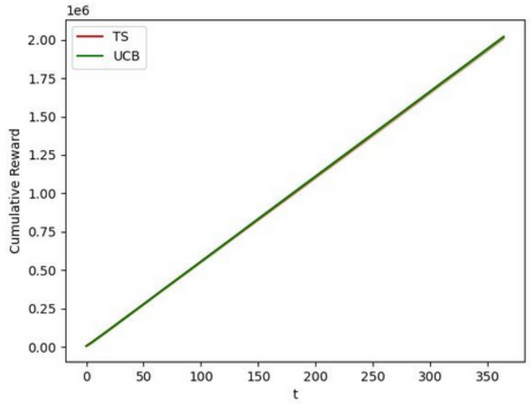
STEP 2: SETUP

- At each round, a price is pulled and the advertising problem is solved with the suggested price

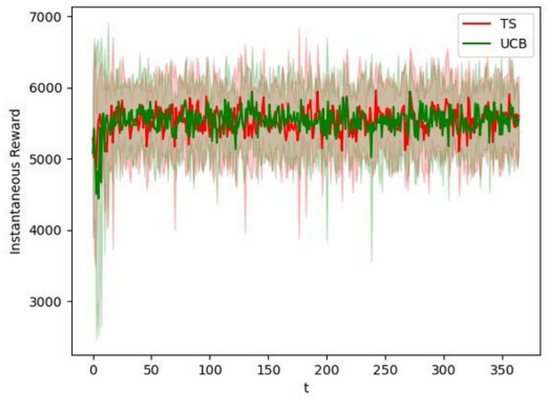
- To average the results, 15 experiments (computationally heavy) were run over 365 days.
- UCB and TS are the two algorithms to compare for the advertising problem

- The Bidding Environment, during each round, generates the number of clicks, the cumulative cost, and the gained reward.

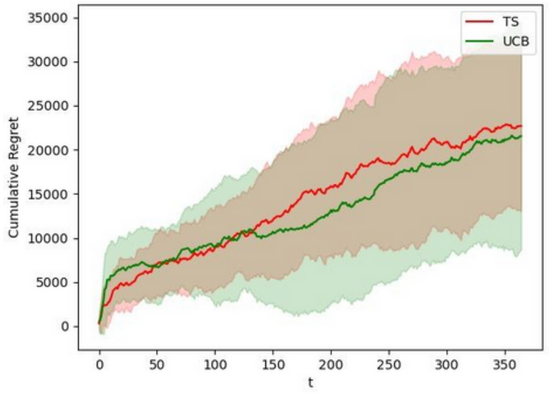
STEP 2: RESULTS



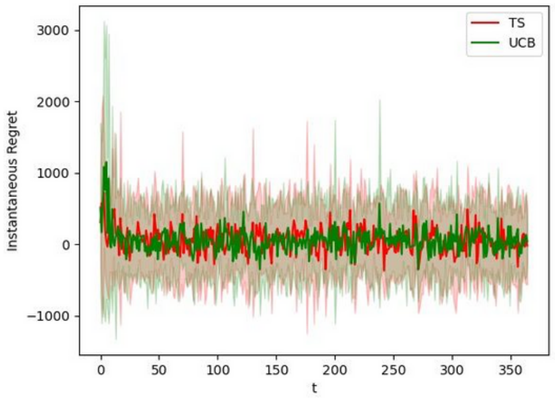
Cumulative reward



Istantaneous reward



Cumulative regret



Istantaneous regret

- A sublinear regret has been achieved
- Both TS and UCB1 get very similar results, but TS performed a little better, maybe because:
  - The optimistic nature or UCB1
  - An unbalanced trade-off between exploration and exploitation: a small number of arms, so no need for an extensive search

POLITECNICO MILANO 1863

STEP 3: LEARNING FOR JOINT PRICING AND ADVERTISING

SCENARIO

ALL USERS BELONG TO CLASS C1

PRICING CURVES ARE UNKNOWN

ADVERTISING CURVES ARE UNKNOWN

REQUEST

TS

APPLY GP-TS ALGORITHM

UCB

APPLY GP-UCB ALGORITHM

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

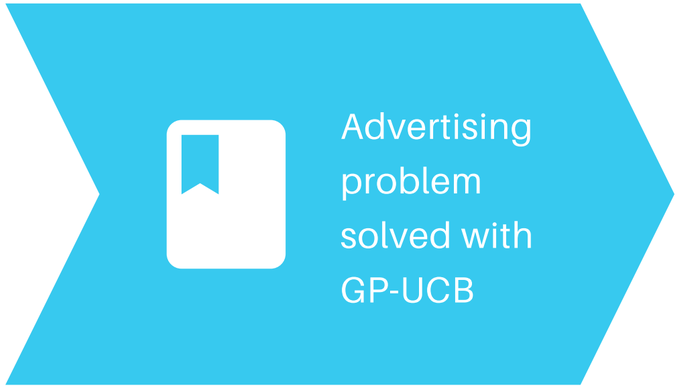
STEP 3: SETUP

- Two phases of optimization: price and advertising
- At each round, a price is pulled and the advertising problem is solved with the suggested price

- To average the results, **50 experiments** are run over 365 days.
- No request has been done over the pricing problem, so the algorithm used is **TS** because it was the one that **performed better in the first step**
- **UCB1 and TS** are the two algorithms to compare for the advertising problem

- The Bidding Environment, during each round, generates the **number of clicks** and the **cost of the bid**.
- The Pricing Environment, during each round, generates the number of **successes**.

STEP 3: Two levels of optimizations

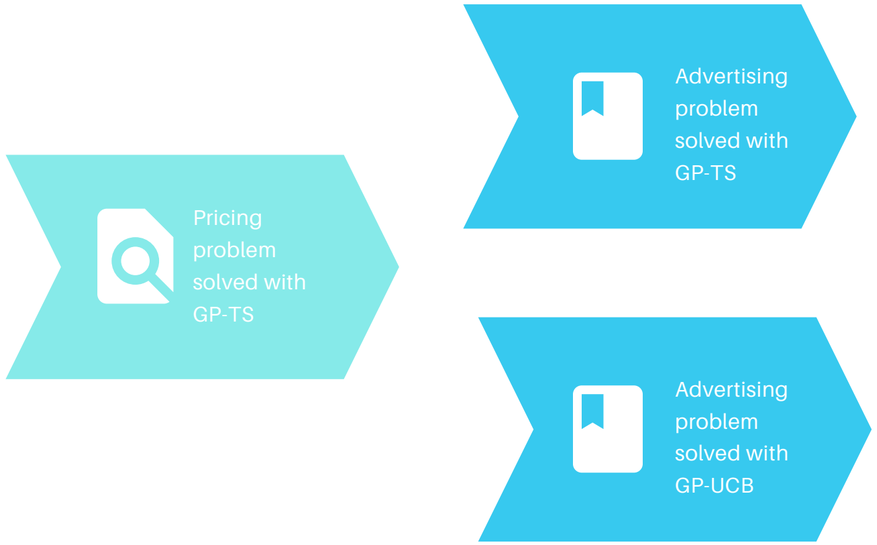


Divide et impera

Once that the price is fixed, so the pricing problem is solved with GP-TS. We can move to the advertising problem.

GP-TS and GP-UCB are compared in the final stage

STEP 3: Two levels of optimizations



Both problems are harder to handle, and the two-level optimization approach brings the following advantages:

- **Easier** to understand (divide et impera approach quite used in computer science problems)
- Brings to an **optimal solution**
- **Computationally less heavy**

POLITECNICO MILANO 1863

STEP 3: Optimization process

Pricing problem

TS chooses the arm with **highest** value from the linked Beta distribution

Optimal price and conversion rate found

The **price** and the **conversion rate** related are given

Advertising problem

Two different Gaussian processes to estimate **#clicks** and **cost** with TS and UCB algorithm

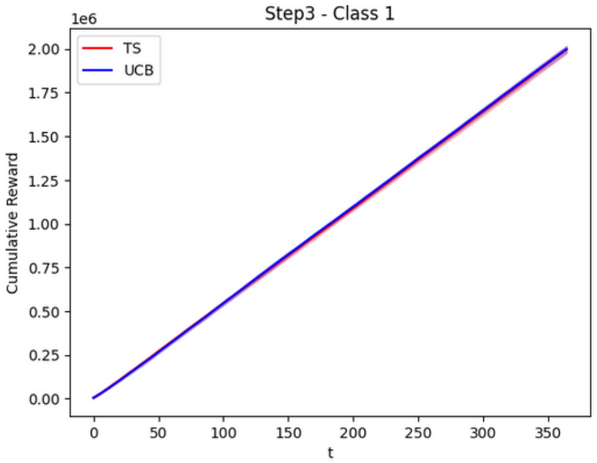
Price and bid obtained

The chosen **price and bids** are implemented

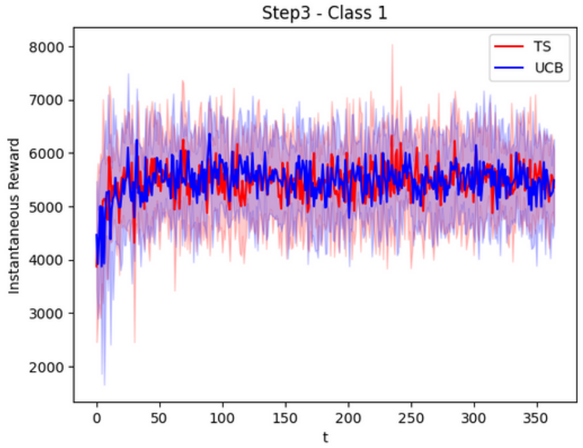
Iterate

The environment gives real results and these are used to to **update** the Ts learner and the to GP of the advertising part

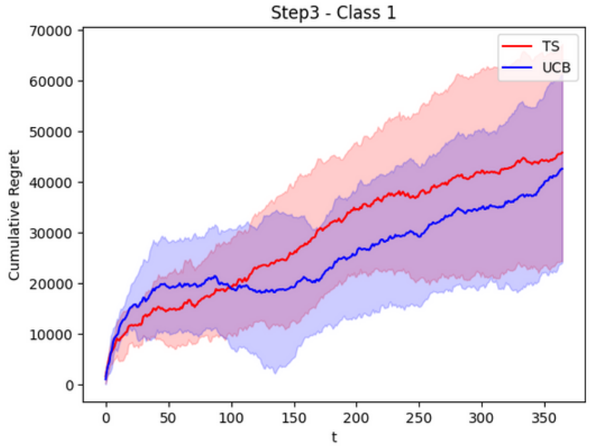
STEP 3: RESULTS



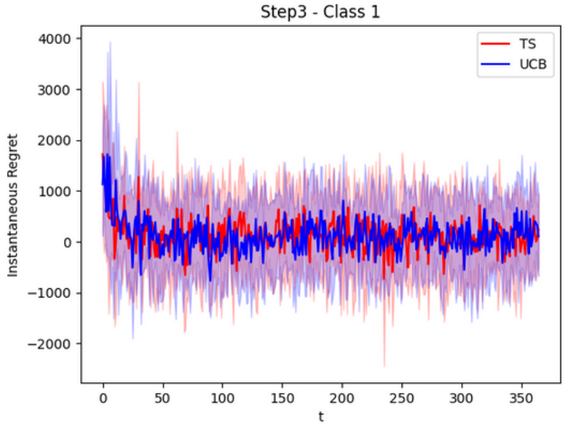
Cumulative reward



Istantaneous reward



Cumulative regret



Istantaneous regret

- A slightly sublinear regret has been achieved
- Both TS and UCB1 get very similar results
- In particular, TS has a lower regret for the first 100 days, but then UCB1 takes the lead
- The reward is a bit higher the reward we had in STEP1

POLITECNICO MILANO 1863

STEP 4: CONTEXTS AND THEIR GENERATION

SETTING

ALL CLASSES OF USERS: C1, C2, C3

PRICING CURVES ARE UNKNOWN

ADVERTISING CURVES ARE UNKNOWN

SCENARIO 1

CONTEXT STRUCTURE IS KNOWN

TS UCB

APPLY GP-TS & GP-UCB ALGORITHMS

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

SCENARIO 2

CONTEXT STRUCTURE IS UNKNOWN

TS UCB

APPLY GP-TS & GP-UCB ALGORITHMS WITH / WITHOUT CONTEXT

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

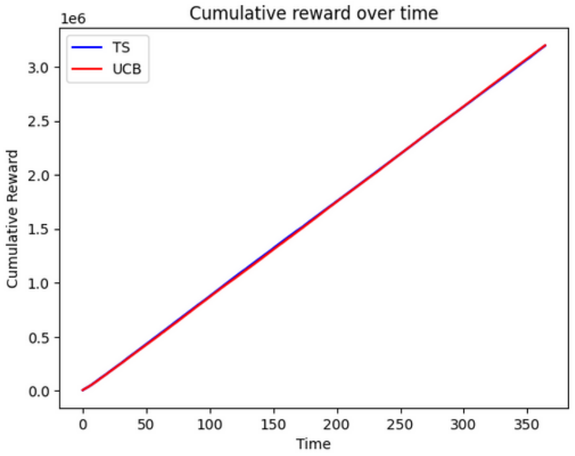
STEP 4.1: SETUP

- We **treat each class independently**, assigning a **different Learner** to each of them
- At each round, for each class, we **pull both a price and a bid**
- Being a disaggregate scenario, we consider the **regret as the sum of each class' regret**

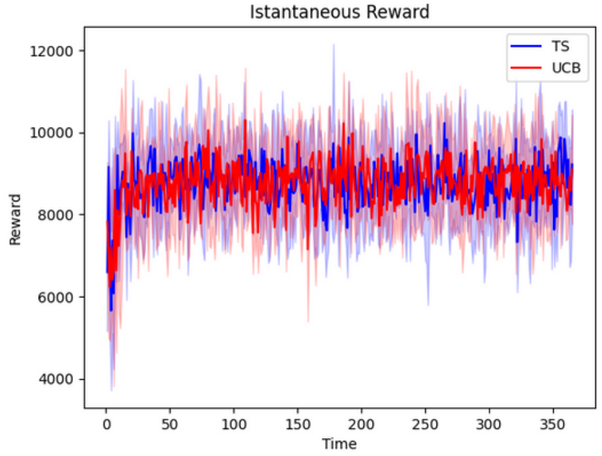
- The execution takes a lot of time
- To average the results feasibly, **only 5 experiments** are run over 365 days.
- UCB1 and TS are the two algorithms we are comparing

- The Environment, during each round, generates the reward based on both the price and the bid that has been pulled by the Learner
- The Learner has to learn both the optimal price and the two advertising curves

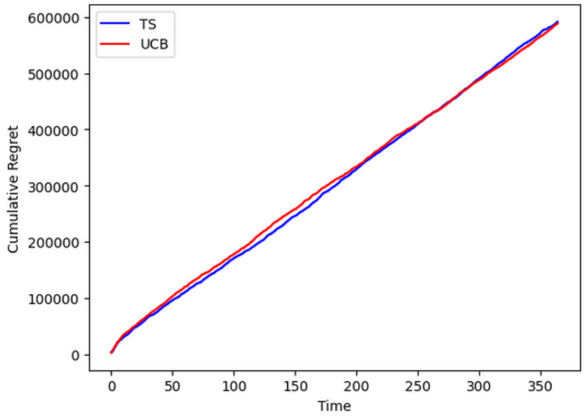
STEP 4.1: RESULTS



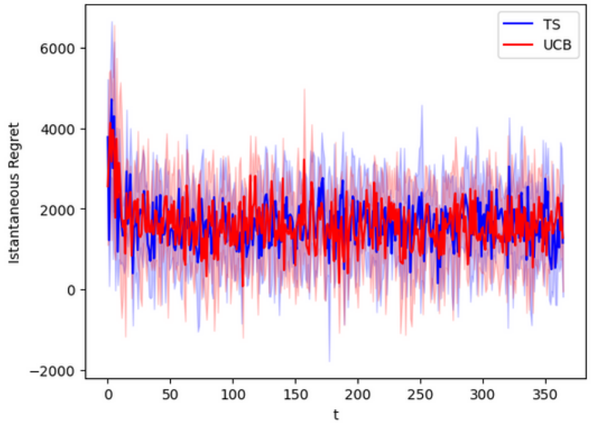
Cumulative reward



Istantaneous reward



Cumulative regret

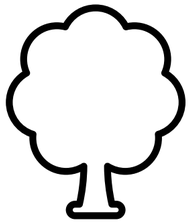


Istantaneous regret

- A slightly sublinear regret has been achieved
- Both TS and UCB1 get very similar results
- In particular, TS has a lower regret for the first 250 days, but then UCB1 takes the lead
- The reward is twice the reward we had in STEP1

STEP 4.2: SETUP

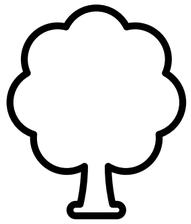
- In this scenario, we adopt the **context-generation algorithm**
- In the beginning, we have a general aggregated model
- **Every two weeks**, the context generation algorithm is performed: **new contexts will be introduced** and some existing contexts could be removed



- **Every possible combination of features is explored:** there are only two binary features
- For each combination, we compute a probabilistic lower bound for the reward that we would get from that context
- For each context, we determine the **best price and the best bid**, considering the two parameters: #clicks and cost

- The Context Manager is responsible for creating and optimizing the context

STEP 4.2: SETUP



- The probabilistic lower bound we use is the Hoeffding's bound, with a confidence of 95%

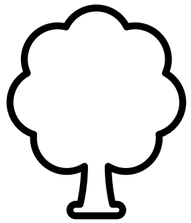
$$\bar{x} - \sqrt{-\frac{\log(\delta)}{2|Z|}}$$

- When a feature is chosen, **we have to decide whether it is worth splitting**. We split when the lower bound of the expected reward is not smaller than the lower bound of the expected reward when we do not split.
- Therefore, the splitting condition is

$$\underline{p}_{c_1}\underline{\mu}_{a^*_{c_1},c_1} + \underline{p}_{c_2}\underline{\mu}_{a^*_{c_2},c_2} \geq \underline{\mu}_{a^*_{c_0},c_0}$$

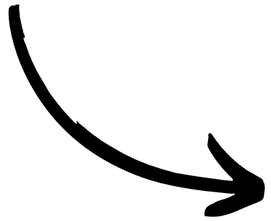
C: the context structure. C0 is the original one, C1 and C2 are the newly generated ones.
p: the lower bound of the probability of occurrence of a user that belongs to the context C
mu: the lower bound of the reward of the optimal arm of context C

STEP 4.2: SETUP



- The confidence interval per a Gaussian variable

$$\overline{x} \pm z\frac{s}{\sqrt{n}}$$



- Lower bound per #clicks

$$\mathrm{CLB} = \frac{\mathrm{clicks}}{t} - 1.96\frac{\sigma_{\mathrm{clicks}}}{\sqrt{t}}$$

- Upper bound per cumulative cost

$$\mathrm{CCUB} = \frac{\mathrm{cost}}{t} + 1.96\frac{\sigma_{\mathrm{cost}}}{\sqrt{t}}$$

STEP 4.2: Computation

- Optimal price

$$\bar{p} = \arg\max_{p} CR(p) \cdot (0.7 \cdot p)$$

- Optimal bid

$$\bar{b} = \arg\max_{p} \frac{1}{CLB_b} \left(CR_{\bar{p}} CLB_b(\bar{p} \cdot 0.7) - CCUB_b\right)$$

- Context reward

$$\underline{\mu}_{a_c^*,c} = \frac{1}{CLB_{\bar{b}}} (CR_{\bar{p}} CLB_b(\bar{p} \cdot 0.7) - CCUB_{\bar{b}})$$

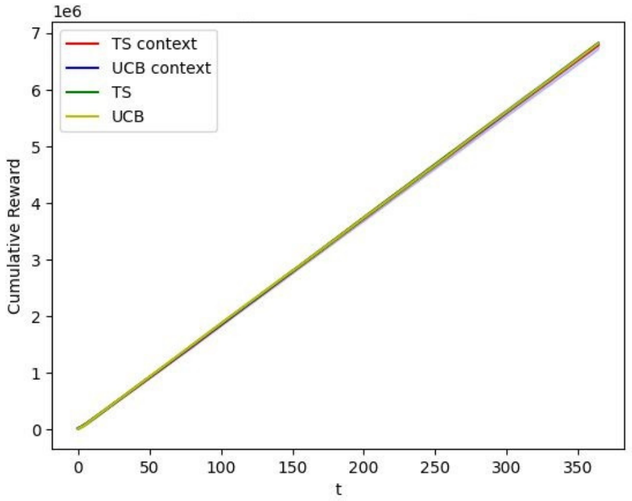
- Context probability

$$\underline{p}_c = \frac{\text{clicks}_c}{\sum_{i \neq c} \text{clicks}_i} - \sqrt{-\frac{\log 0.95}{2\text{clicks}_c}}$$

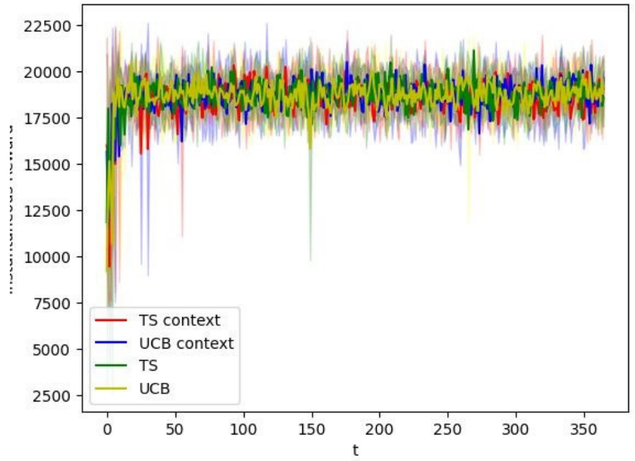
Note that when contexts are computed, each one is assigned a learner. For already existing contexts, the learner is the same

POLITECNICO MILANO 1863

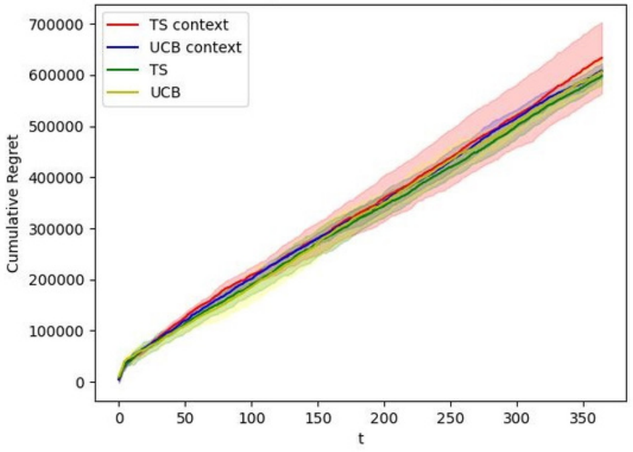
STEP 4.2: RESULTS



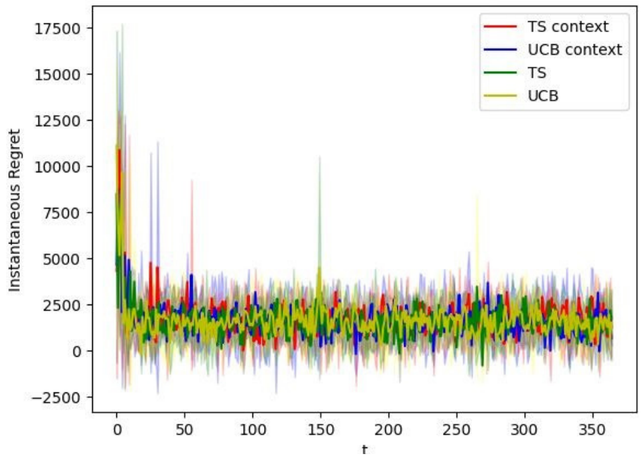
Cumulative reward



Instantaneous reward



Cumulative regret



Instantaneous regret

- A **sublinear** regret has been achieved
- The **cumulative reward for the deployed algorithms is similar**
- Surprisingly, the **contextual** version of TS achieves the highest cumulative regret

- The reward is **5 times** the reward we had in STEP1
- However, **the regret is larger than in scenario 1**. This makes sense because the context structure is unknown
- We **expected to see more differences** between the contextual and the non-contextual versions, but **the results are very similar**

POLITECNICO MILANO 1863

STEP 5: DEALING WITH NON-STATIONARY ENVIRONMENTS WITH TWO ABRUPT CHANGES

SCENARO

ALL USERS BELONG TO CLASS C1

PRICING CURVES ARE UNKNOWN, NON-STATIONARY AND WITH THREE DIFFERENT PHASES

ADVERTISING CURVES ARE KNOWN

REQUEST

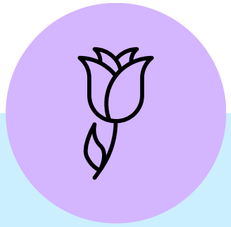
APPLY UCB1 ALGORITHM WITH SLIDING WINDOWS

APPLY UCB1 ALGORITHM WITH CHANGE DETECTION

SENSITIVITY ANALYSIS

PLOT CUMULATIVE & INSTANTANEOUS REWARD / CUMULATIVE & INSTANTANEOUS REGRET

3 PHASES



SPRING

Regular behaviours

Best price: 150€

CHRISTMAS

Increased price for gift

Best price: 200€

HOLIDAY SALES

Just real perceived deals

Best price: 100€

POLITECNICO MILANO 1863

3 LEARNERS

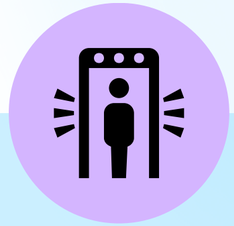
Normal UCB

Normal Upper Confidence Bound algorithm

Sliding Window UCB

Parameter proportional to $\sqrt{T}$

More weight to recent observations while still considering a reasonable amount of historical data



Change Detection UCB (CUSUM)

M, ϵ constant

consistent and reliable change detection behaviour

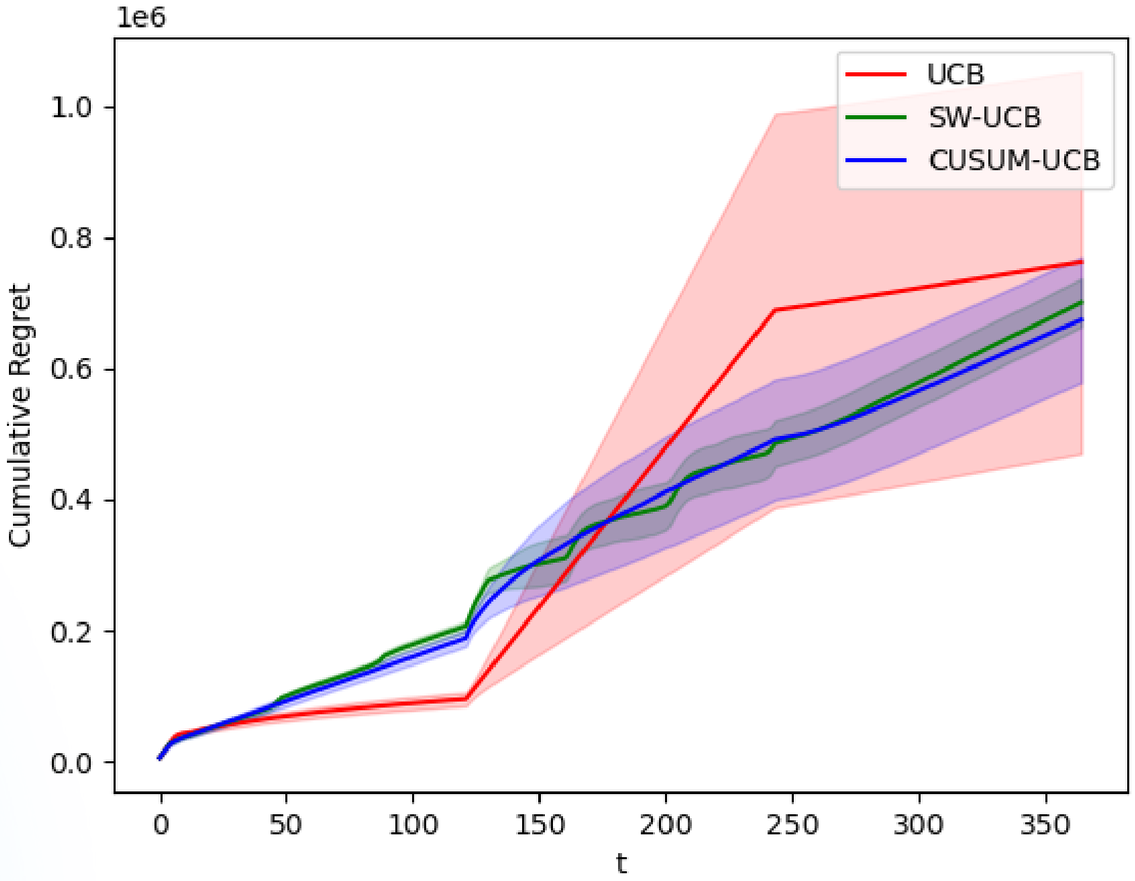
$h \propto \log T$

balance between detecting changes promptly and avoiding false positives

$h \propto \frac{\sqrt{\log T}}{T}$

balance between exploration and exploitation

RESULTS



1. ***CUSUM UCB***
2. ***SLIDING WINDOW UCB***
3. ***UCB***

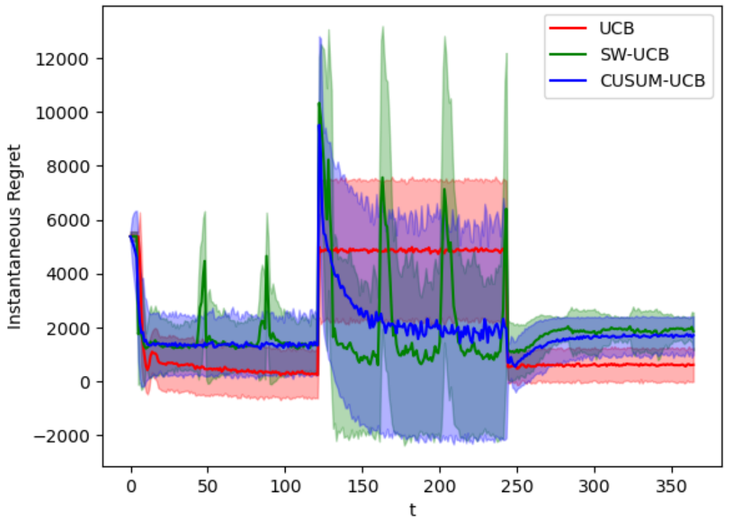
SW-UCB shows peaks that follow the flow of the window

1. Sample exit window -> exploration (infinite confidence)
2. Worst arm pulled once
3. Poor reward
4. Never pulled again per window

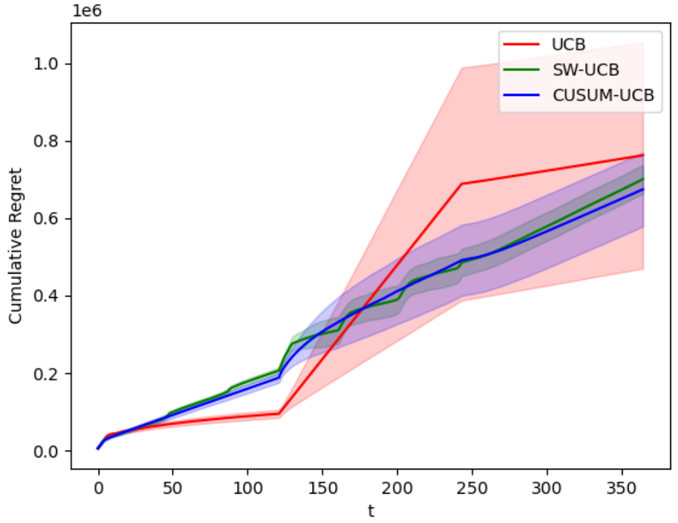
RESULTS

From the considerations about the regret follow the plots about rewards

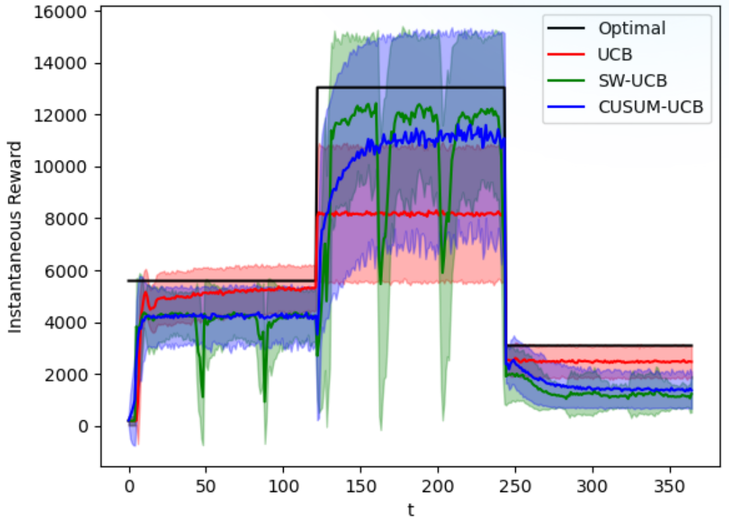
Instantaneous Regret



Cumulative Reward



Instantaneous Reward



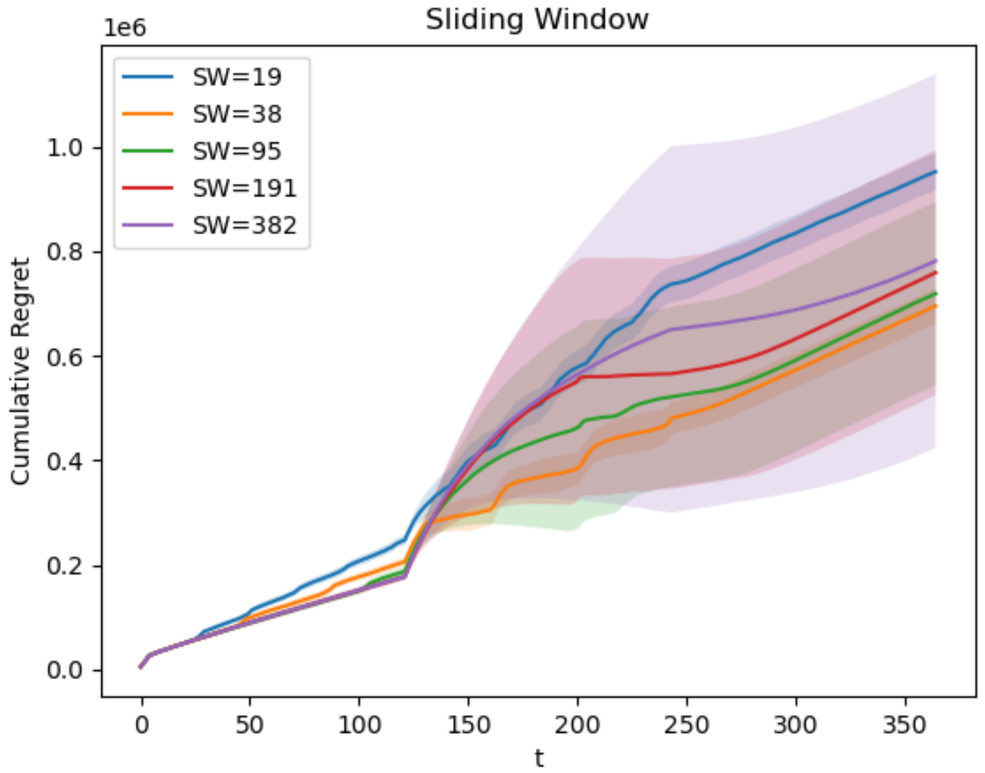
SENSITIVITY ANALYSIS - SOFTWARE PARAMETERS

Sliding Window Size

It determines the number of recent observations or time steps considered. It is critical to the analysis's responsiveness, stability, noise sensitivity, and computing complexity.

SENSITIVITY ANALYSIS - SLIDING WINDOW

$$window\_sizes = \{1, 2, 5, 10, 20\} * \sqrt{T}$$



- responds quickly to short-term changes
- more sensitive to abrupt shifts or anomalies
- more subject to noise and random oscillations
- potentially less trustworthy estimations

- smoothed and steady representation of the data
- limiting the impact of short-term volatility
- removal of random noise and outliers
- response to immediate changes is diminished

Medium sliding window sizes perform better then the largest and smallest ones in terms of regrets

SENSITIVITY ANALYSIS - CUSUM PARAMETERS

M

Sensitivity Parameter

The magnitude threshold used for change detection. It determines the number of consecutive observations needed to trigger a change detection event.

eps

Threshold for Change Detection

The threshold for detecting a change in the observed data. It determines the minimum magnitude of change required to trigger a change detection event.

h

Confirmation Threshold

The decision threshold for change detection. It determines the level of deviation required to trigger a change detection event.

alpha

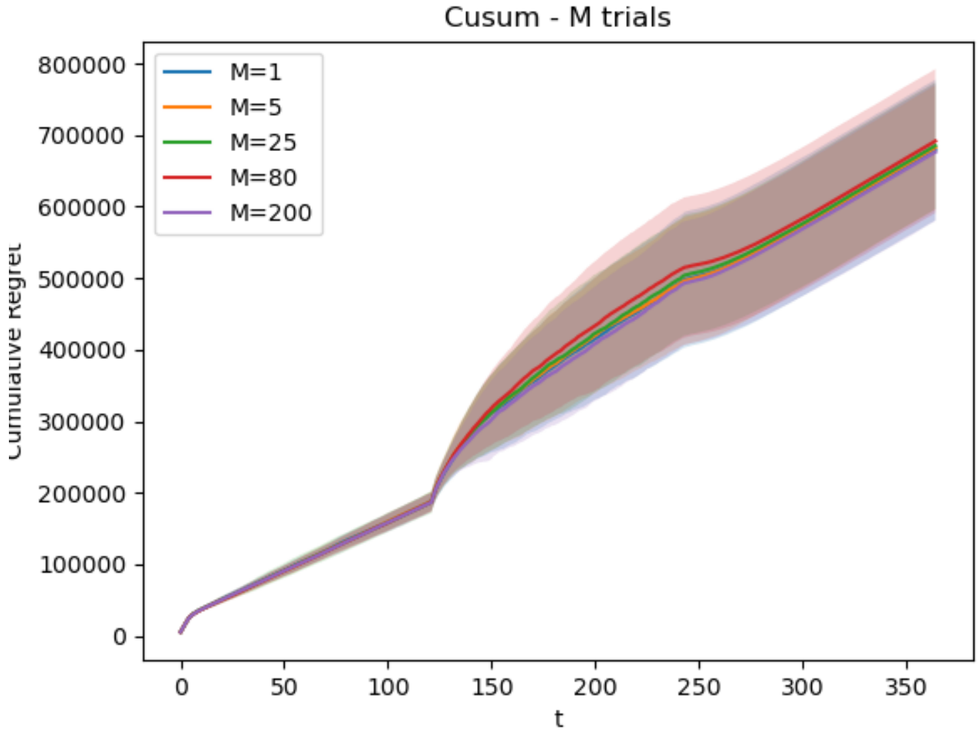
Exploration Probability

The significance level or the type I error rate. It determines the threshold for considering a change as statistically significant.

SENSITIVITY ANALYSIS - CUSUM (M)

$$M = \{1, 5, 25, 80, 200\}$$

Sensitivity Parameter



- more sensitive to small changes
- detect subtle variations quickly
- rapid adaptation to changes
- increase the possibility of false-positive detections

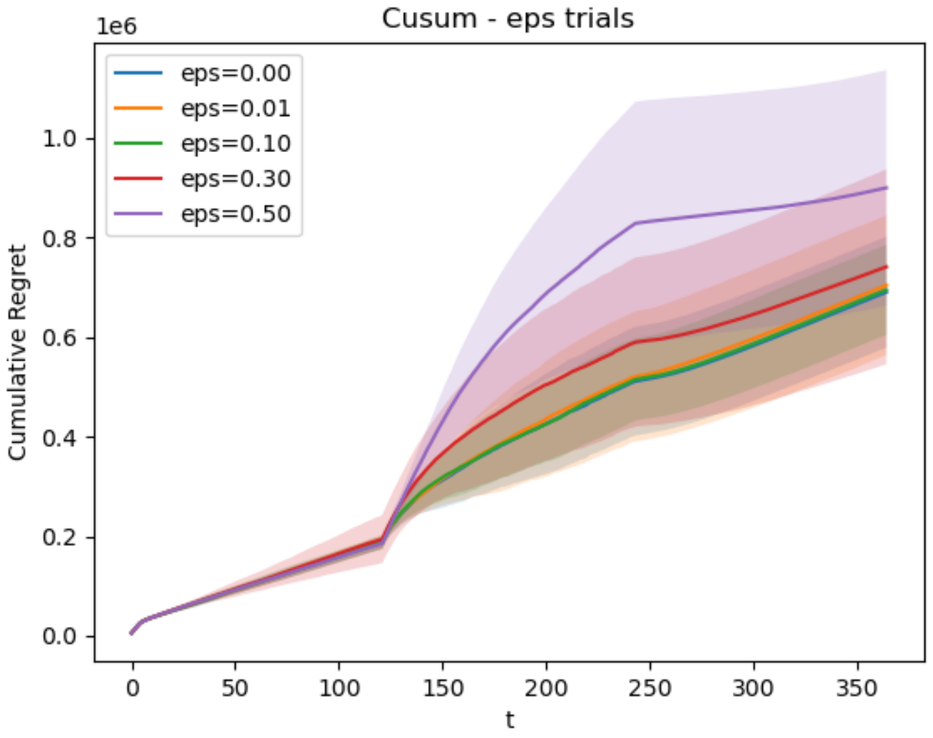
- needing a greater magnitude of change to trigger a detection
- minimises the likelihood of false-positive detections
- slower adaptability
- potentially ignoring smaller changes

No much difference between the different values. M as a parameter has less relevance and importance than others

SENSITIVITY ANALYSIS - CUSUM (eps)

$$\epsilon = \{0.001, 0.01, 0.1, 0.3, 0.5\}$$

Threshold for Change Detection



- sensitive to even minor changes in the data
- high responsiveness to identify tiny shifts
- may increase the risk of false-positive detections
- minor variations may misidentified as substantial changes

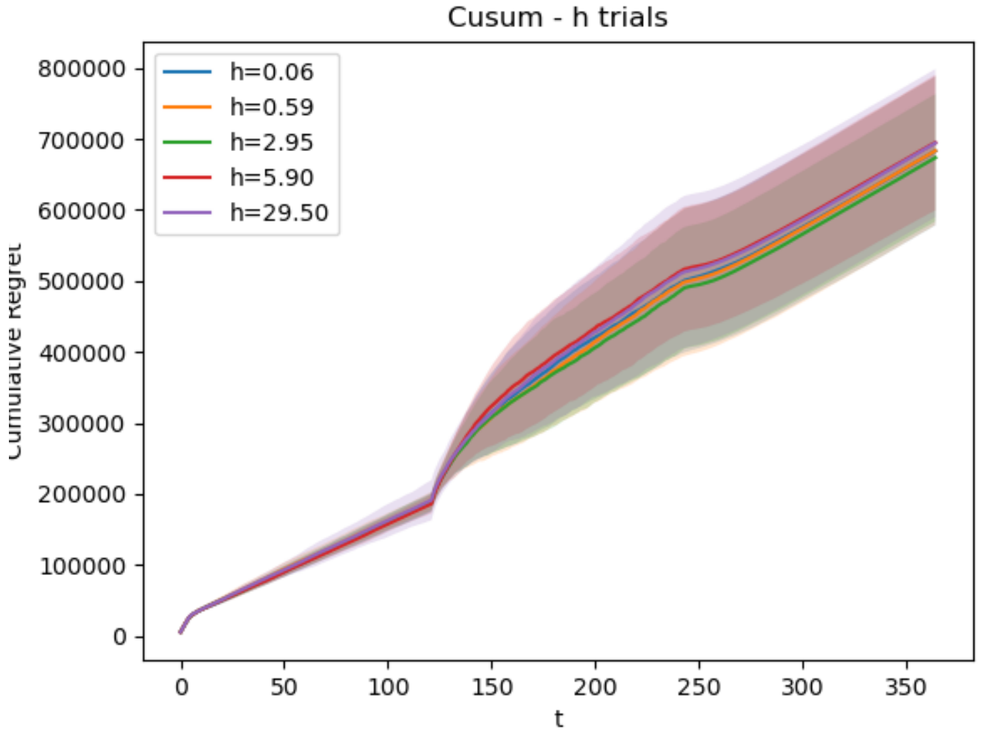
- needing a greater magnitude of change to trigger a detection
- minimises the likelihood of false-positive detections
- reduce the sensitivity to minor changes
- potentially overlooking smaller changes

An higher regret is obtained from higher epsilon values, with a decreasing regret for smaller values.

SENSITIVITY ANALYSIS - CUSUM (h)

$$h = \{0.01, 0.1, 0.5, 1, 5\} * \log T$$

Confirmation Threshold



- extremely sensitive to departures from expected behaviour
- detects tiny changes in observed data fast
- increase the likelihood of false-positive detections
- random fluctuations may detected as substantial changes

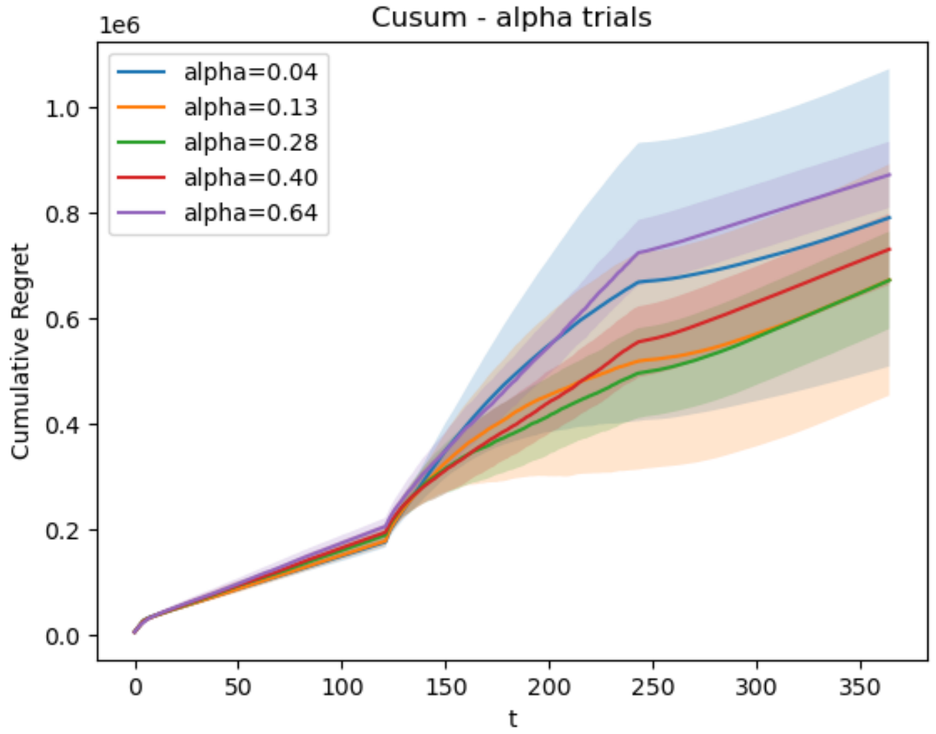
- needing a higher amount of deviation to trigger a detection
- minimises the likelihood of false-positive detections
- reduce the sensitivity to minor changes
- detecting significant and noteworthy deviations is more critical

The values of h are all almost similar, underlining a rather marginal relevance of the parameter.

SENSITIVITY ANALYSIS - CUSUM (alpha)

$$\alpha = \{0.1, 1, 5, 10, 25\} * \sqrt{\frac{\log(T)}{T}}$$

Exploration Probability



- stronger proof is required before a change
- minimising the possibility of false-positive detections
- reduce the sensitivity
- potentially causing to overlook certain true changes in the data

- more sensitive at identifying possible changes
- increases the possibility of false-positive detections
- trade-off between sensitivity and specificity

The best values are those in the middle. The exploration part works well when balanced, so if you explore too little, this leads to less convergence, as well as if you explore too much at random.

POLITECNICO MILANO 1863

STEP 6: DEALING WITH NON-STATIONARY ENVIRONMENTS WITH MANY ABRUPT CHANGES

SETTING

ALL USERS BELONG TO CLASS C1

PRICING CURVES ARE UNKNOWN, NON-STATIONARY AND WITH THREE DIFFERENT PHASES

ADVERTISING CURVES ARE KNOWN

SCENARIO 1

ENVIRONMENT IS NON-STATIONARY + FIXED BID

DEVELOP & APPLY EXP3 ALGORITHM

VERIFY THAT EXP3 PERFORMS WORSE THAN UCB1 NON-STATIONARY

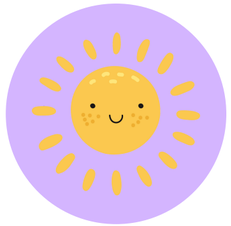
SCENARIO 2

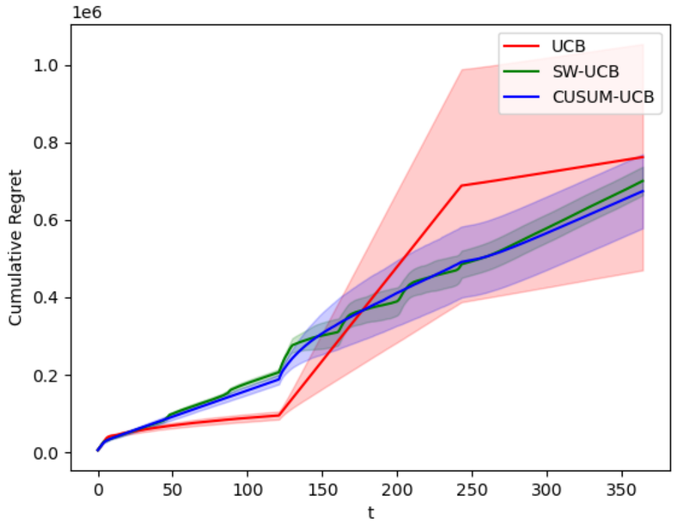
ENVIRONMENT HAS AN HIGHER DEGREE OF NON-STATIONARITY

APPLY EXP3, UCB1 & THE TWO NON-STATIONARY FLAVORS OF UCB1

VERIFY THAT EXP3 OUTPERFORMS THE NON-STATIONARY VERSION OF UCB1

3 PHASES: GETTING BETTER





Step 5 results

- We are now going to simplify the previous scenario while trying to improve our results
- In this new setting, the bid is fixed. We choose the optimal bid value (1.25), computed using the Clairvoyant Algorithm
- Moreover, we introduce a new algorithm that should perform better in non-stationary environments

POLITECNICO MILANO 1863

Introducing: EXP3

- EXP3 is designed to work for MAB problems
- It actively explores the arms, to improve its knowledge about them
- The exploration is based on a probability distribution over the arms ("action probability")
- At each time step, the pulled arm is chosen upon the action probability

$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{k} w_j(t)} + \frac{\gamma}{N}$$

N: number of arms

Gamma: exploration (hyper)parameter [0,1]

w(t): weights associated to each arm at a given time step

- The weight is computed as

$$w_{j_t}(t+1) = w_{j_t}(t) \cdot e^{\frac{\gamma}{N}\hat{x}_{i_t}}$$

j: arm

x: expected reward from that arm

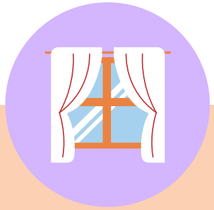
- Due to its exploratory nature, it should be able to adapt better to abrupt changes

POLITECNICO MILANO 1863

4 LEARNERS

Normal UCB

Normal Upper Confidence Bound algorithm



Sliding Window UCB

Parameter proportional to $\sqrt{T}$

More weight to recent observations while still considering a reasonable amount of historical data

Change Detection UCB (CUSUM)

M, ϵ constant

consistent and reliable change detection behaviour

$$h \propto \log T$$

balance between detecting changes promptly and avoiding false positives

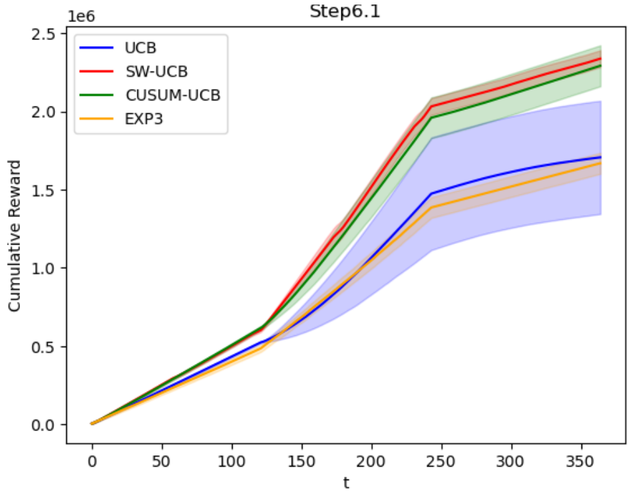
$$h \propto \frac{\sqrt{\log T}}{T}$$

balance between exploration and exploitation

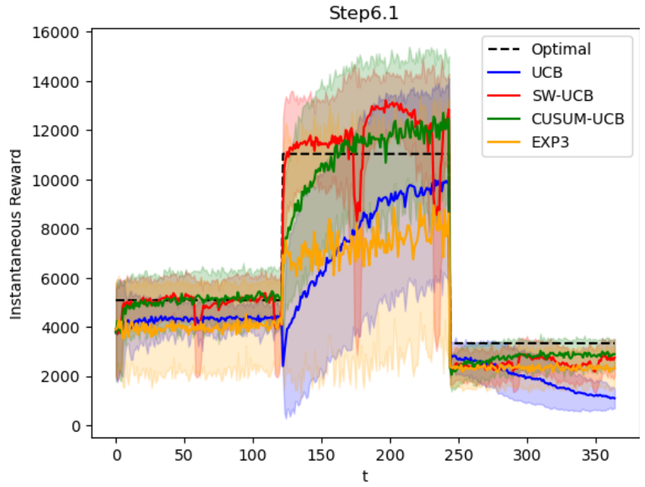
EXP3

Exponential-weight algorithm for Exploration and Exploitation

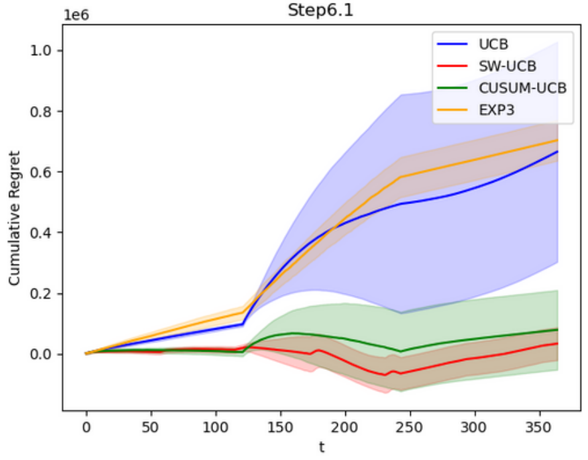
STEP 6.1: RESULTS



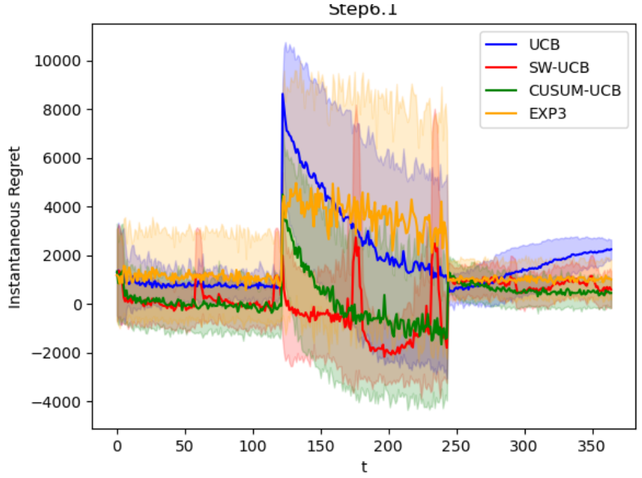
Cumulative reward



Instantaneous reward



Cumulative regret



Instantaneous regret

- EXP3 got the worst performance out of these algorithms, even by tuning its gamma parameter.
- If we compare the instantaneous rewards for UCB1 and EXP3 it is evident that UCB1 learns way faster
- EXP3's weights have been initialized uniformly. Other initializations could help improve the algorithm's performance
- This behavior can be explained by the fact that EXP3 explores the environment a lot and improvements can be seen when the changes are more frequent
- SW-UCB imposes itself as the winner of Step 6.1

3 PHASES: GETTING BETTER

- We said that EXP3 is really about exploration
- What if the environment changes much faster?

5 PHASES

NEW YEAR

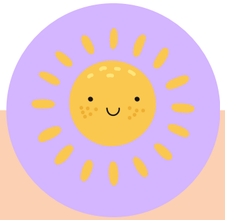
Regular behaviours

Best price: 150€

SPRING

Lower conversion rates

Best price: 250€



SUMMER

Way too hot

Best price: 150€

AUTUMN

Sales!

Best price: 100€

CHRISTMAS

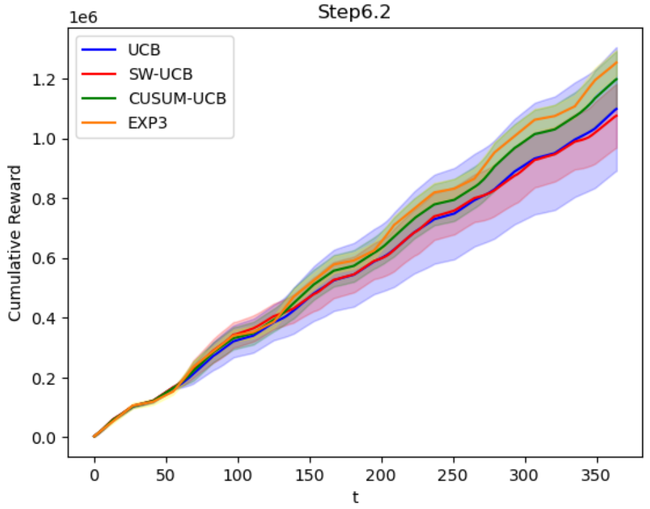
Higher prices due to gifts

Best price: 200€

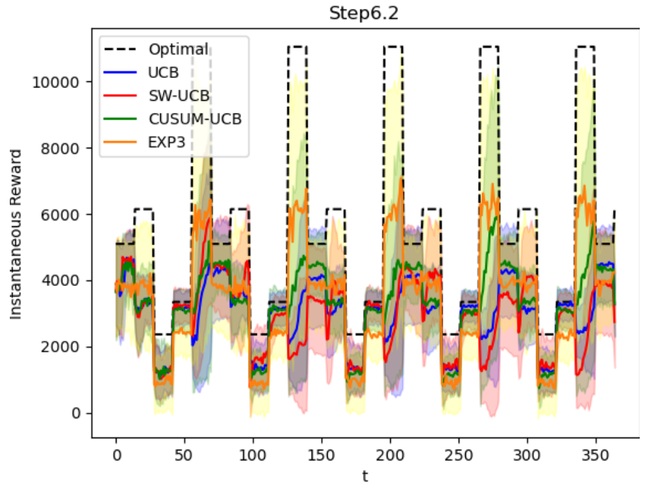
New Year, Sales-Season and Christmas use the same conversion rates used in Step5.

POLITECNICO MILANO 1863

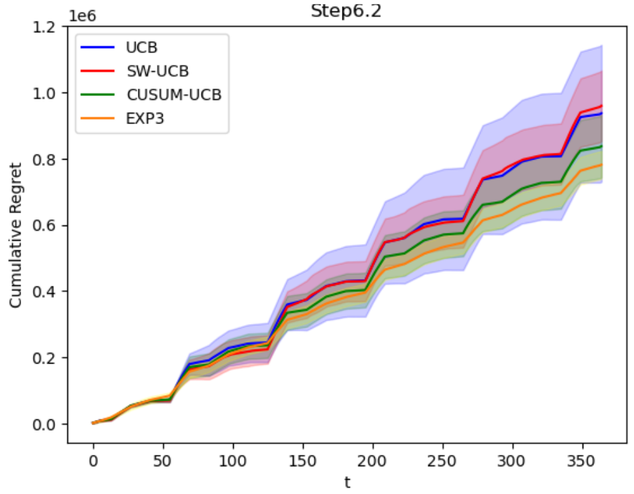
STEP 6.2: RESULTS



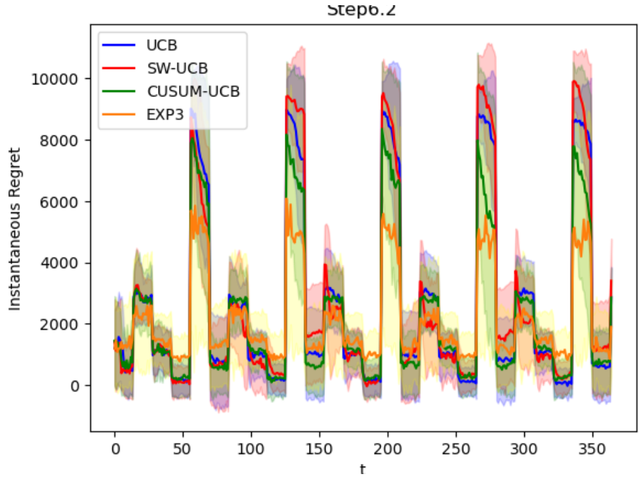
Cumulative reward



Instantaneous reward



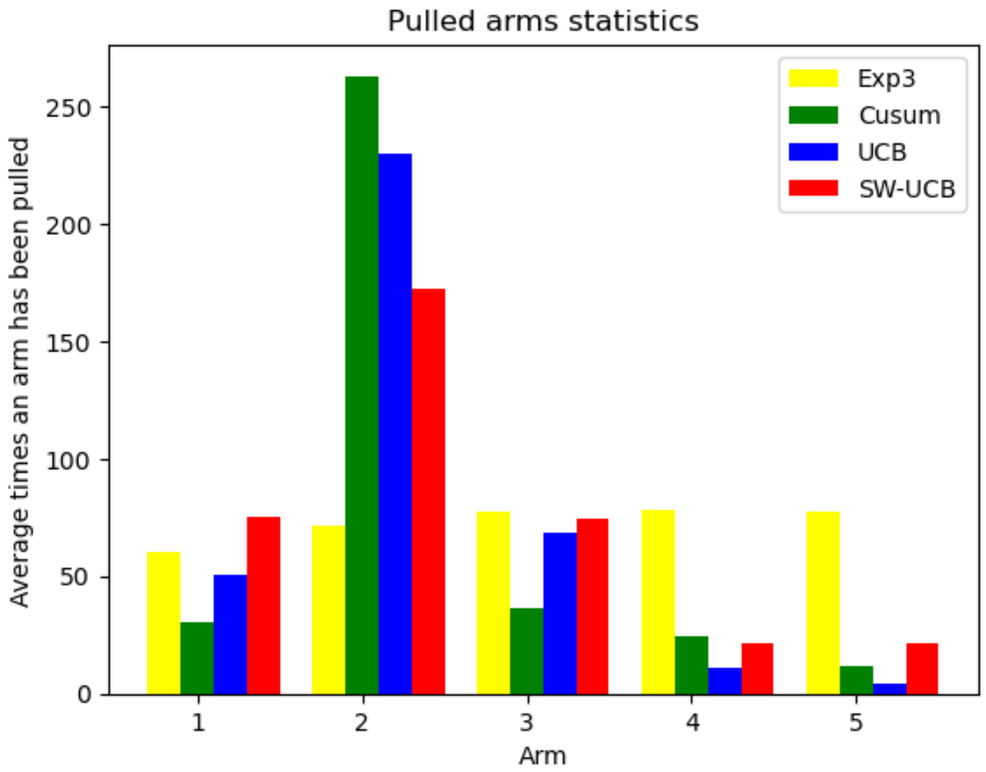
Cumulative regret



Instantaneous regret

- EXP3 brought us the highest cumulative reward and the lowest cumulative regret, implying that it correctly balanced exploration and exploitation.
- EXP3's gamma has been manually tuned very few times: there is a margin for improvement
- CumSumUCB has similar performances w.r.t. EXP3, while the other two algorithms could not achieve optimal results
- The various abrupt changes have decreased the overall cumulative reward w.r.t to the previous scenario

STEP 6.2: ABOUT THE EXPLORATION



- EXP3 has pulled the arms almost equally
- The opposite happened for CumSumUCB, for which arm 2 is definitely the most pulled one

THANK YOU!

Valeria Amato - 10641790
Beatrice Insalata - 10708628
Emanuele Paci - 10681377
Matteo Pancini - 10656944
Andrea Riboni - 10699906

eit Digital

POLITECNICO MILANO 1863