Online Learning Applications

Pricing and Advertising

A.Y. 2022-2023

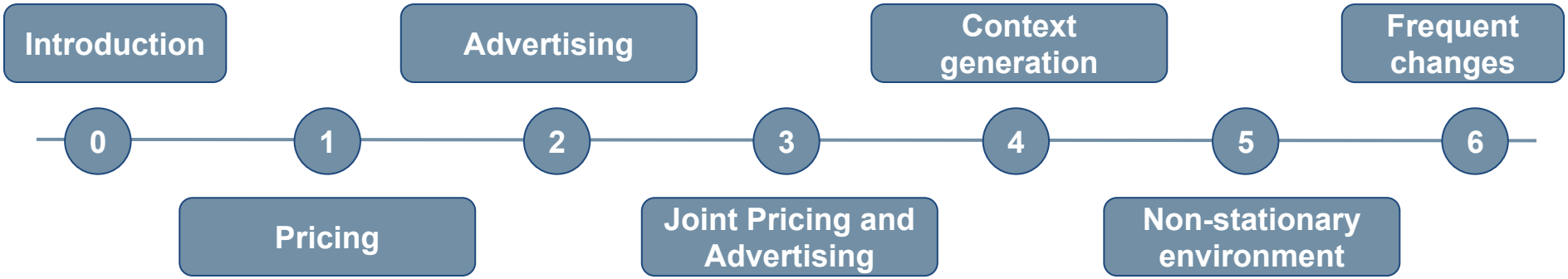
Professors Castiglioni Matteo, Gatti Nicola, Bernasconi de Luca Martino

POLITECNICO
MILANO 1863

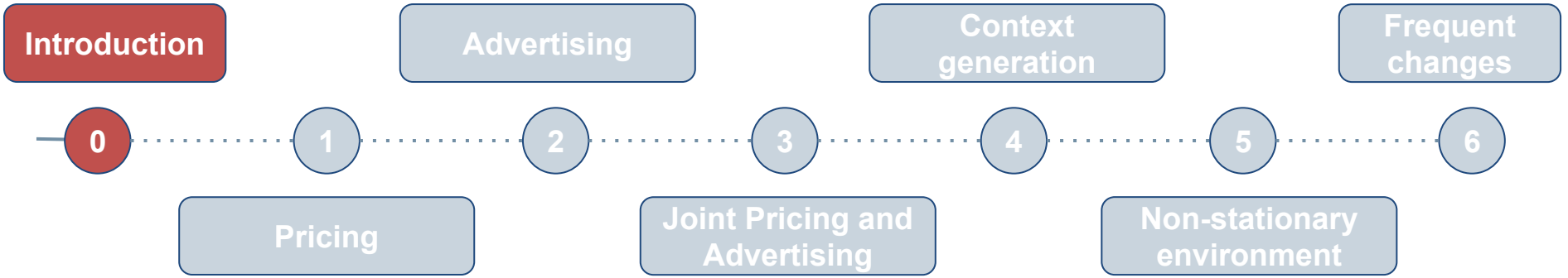
Pertino Paolo	**10729600**
Pesce Leonardo	**10659489**
Sandri Alberto	**10698469**
Simionato Enrico	**10698193**
Vitali Michael	**10730463**

Professors Castiglioni Matteo, Gatti Nicola, Bernasconi de Luca Martino

Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 0: Motivations and environment design
Introducing iPear: e-commerce smartphone debut

e-commerce

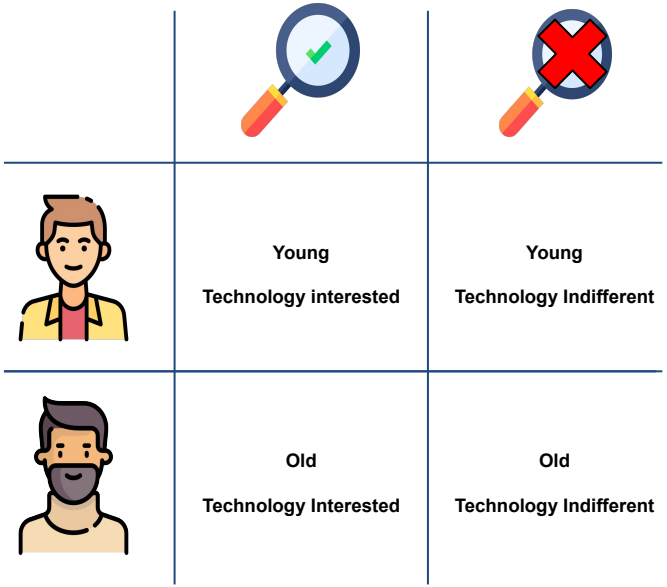
pricing & advertising

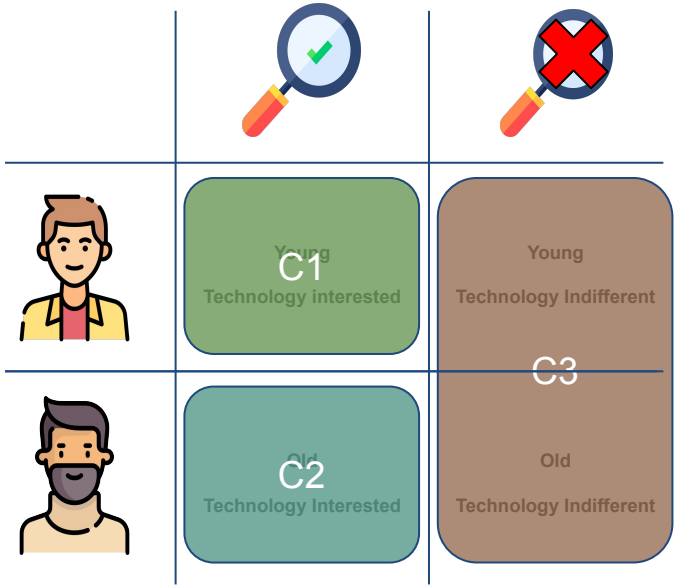
iPear

newest smartphone

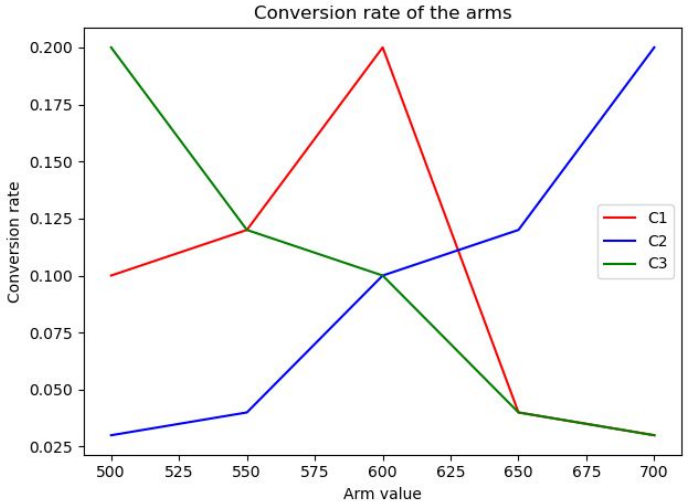
middle-market segment

Step 0: Motivations and environment design
iPear - Users' classes

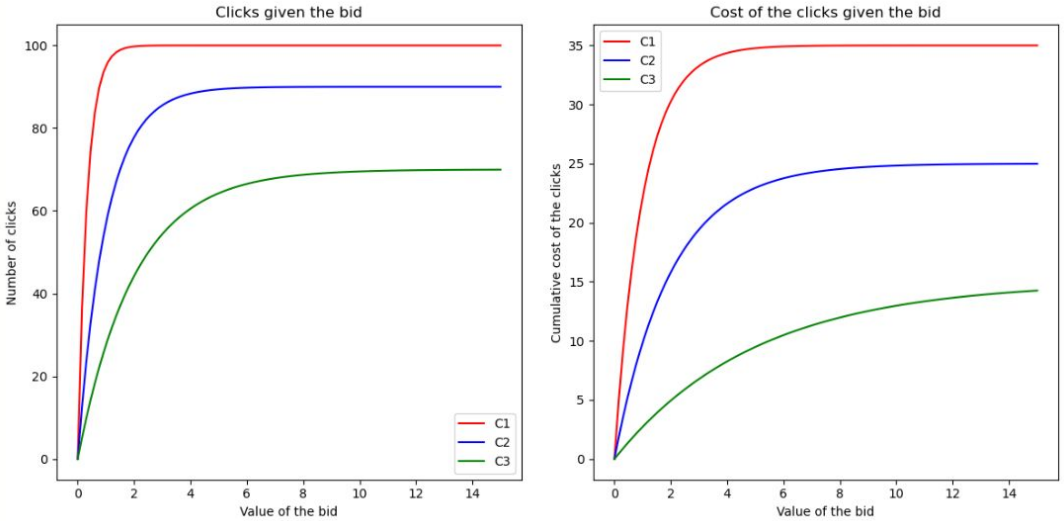




Step 0: Motivations and environment design
iPear - Users' classes - Technicalities



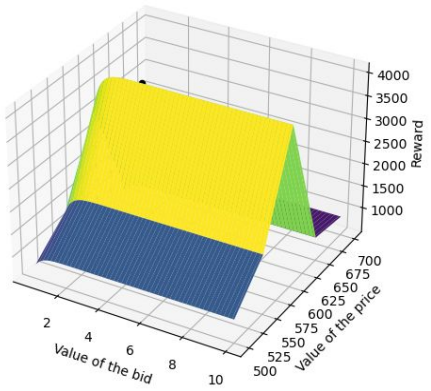
[A1] Conversion probabilities for the three classes



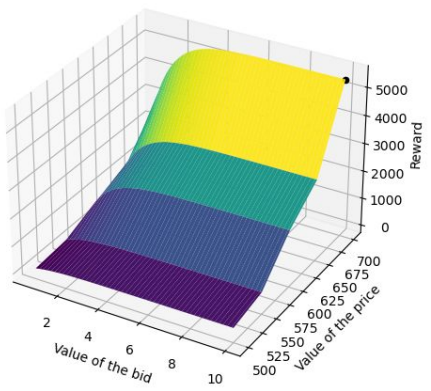
[A2] Clicks and cumulative daily costs curves

Step 0: Motivations and environment design
iPear - Users' classes - Technicalities cont.

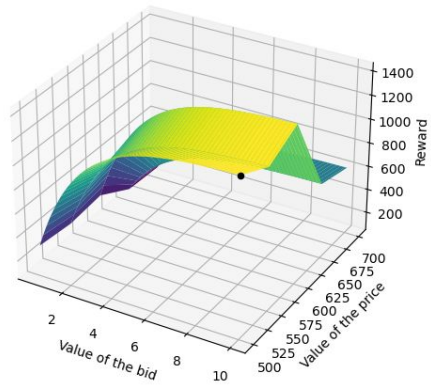
Reward given price and bid for the user category C1



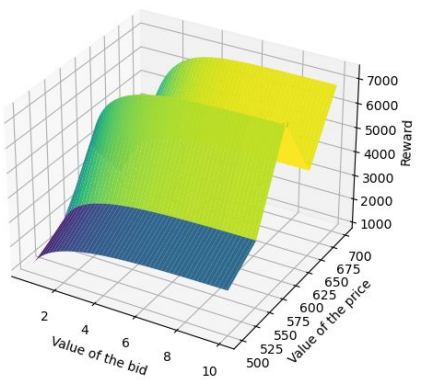
Reward given price and bid for the user category C2



Reward given price and bid for the user category C3



Reward given price and bid using the aggregate model



Step 0: Motivations and environment design
iPear - Clairvoyant

$$reward = \#_{clicks} * \underbrace{\text{conversion probability} * (price - \text{other costs})}_{\textit{conversion\_times\_margin}} - \text{cumulative daily costs}$$

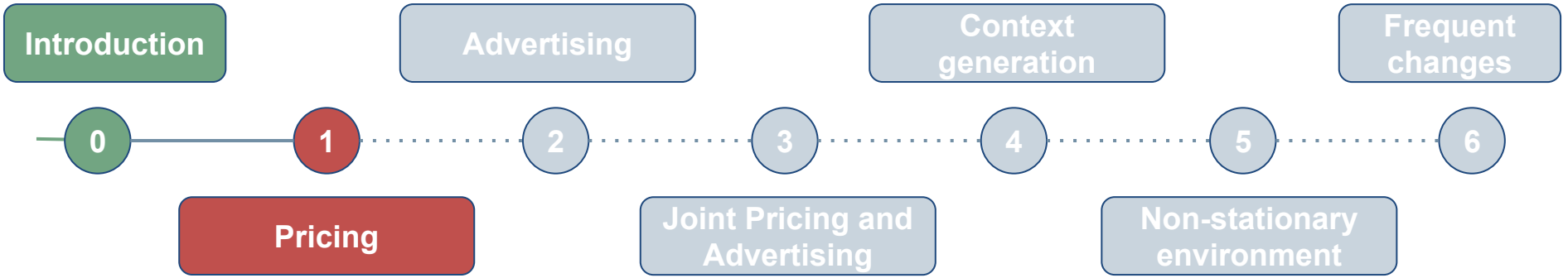
This algorithm works as follows:

***Input**: category*

1. *best_price, conversion_times_margin ← maximization of reward from price given category*
2. *best_bid, reward ← maximization of reward from bid given category and conversion_times_margin*

***Output**: best_price, best_bid, reward*

Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 1: Learning for pricing

Setting

Scenario:

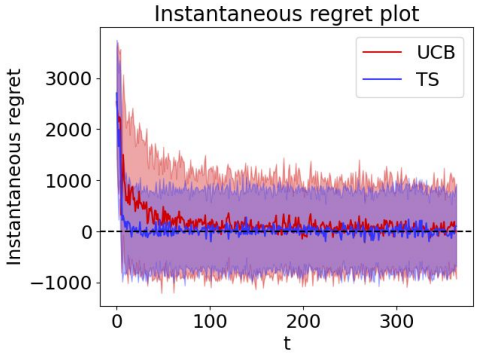
- All the users belong to **class C1**
- Curves related to the **advertising part** are **known**
- Curve related to the **pricing part** is **unknown**

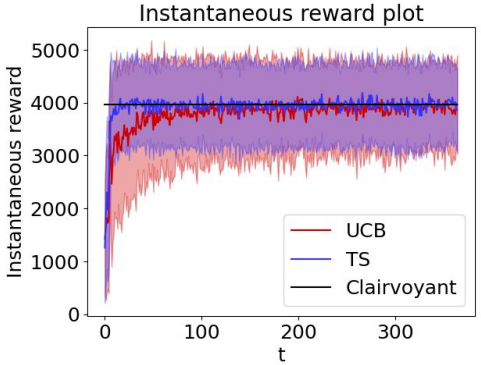
TS: $(\alpha_{a_{t+1}}, \beta_{a_{t+1}}) \leftarrow (\alpha_{a_t}, \beta_{a_t}) + (c_{1,t}, c_{0,t})$

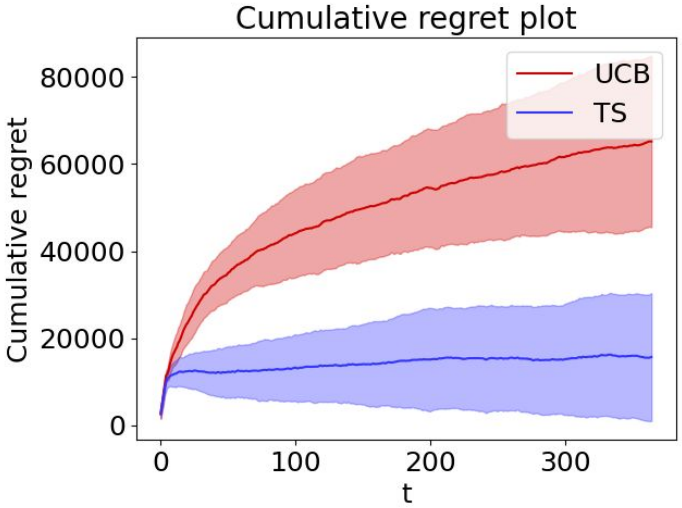
UCB: $n_a(t) \leftarrow n_a(t-1) + c_{1,t} + c_{0,t}$

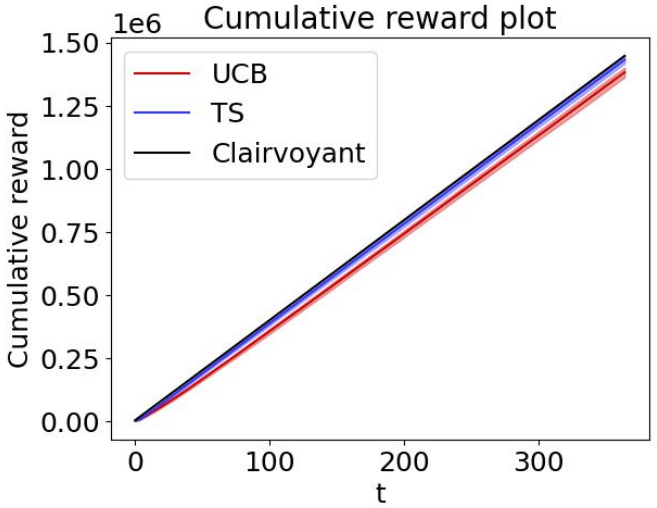
Step 1: Learning for pricing

Results

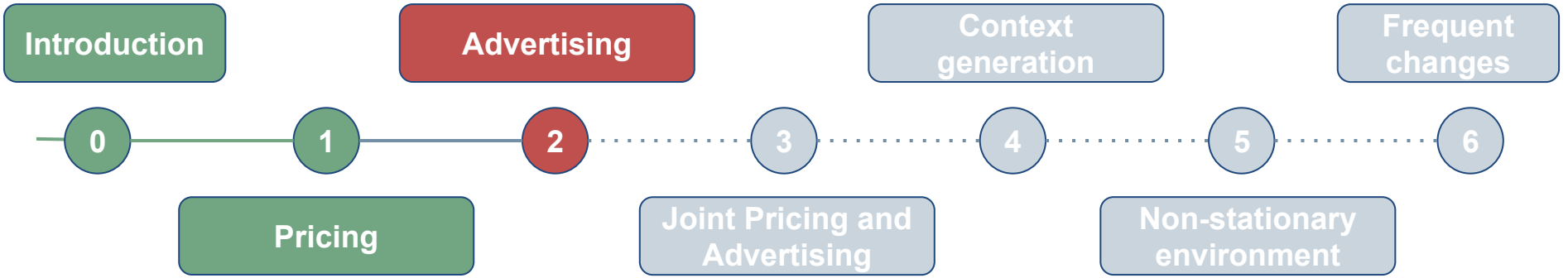








Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

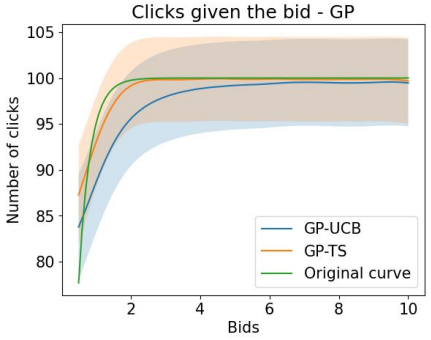
Step 2: Learning for advertising

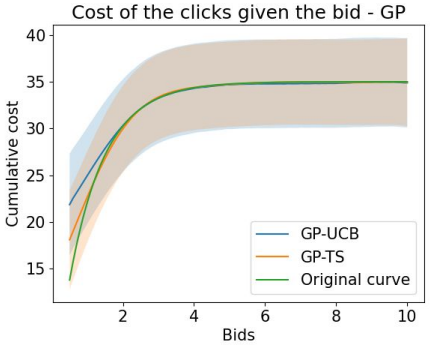
Setting & Assumptions

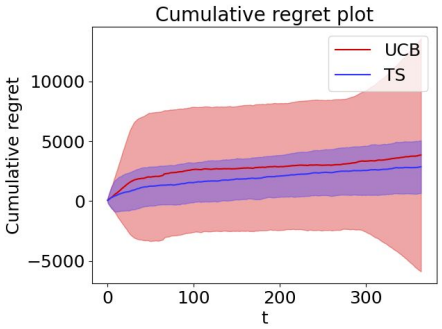
- **Pricing** curve is **known** and considered fixed so the best price is used by maximizing the **optimal conversion rate times margin** term
- **Maximize rewards** while **estimating** two curves: one for the **number of clicks** and one for **cumulative costs** as the **bid varies**
- **Gaussian Processes (GP)** are used for these estimates due to their **ability** to **handle regression problems** with **minimal assumptions** on the **function to estimate**
- **GPs** not only **provide** an **average value** based on observations but also a **measure of uncertainty in** the **regression result**, **represented as** a **probability distribution**
- It's **assumed** that there is **correlation between bids that are close** to each other, **requiring** the **two curves** to be **sufficiently smooth**

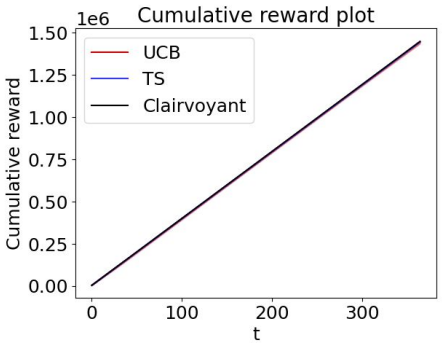
Step 2: Learning for advertising

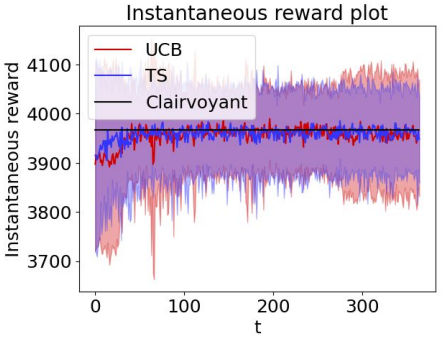
Results

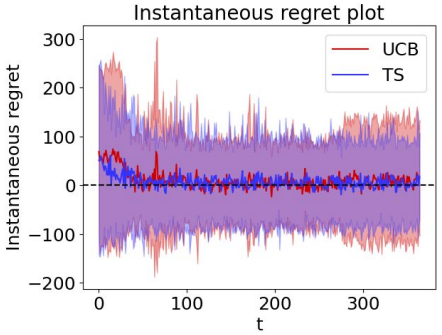




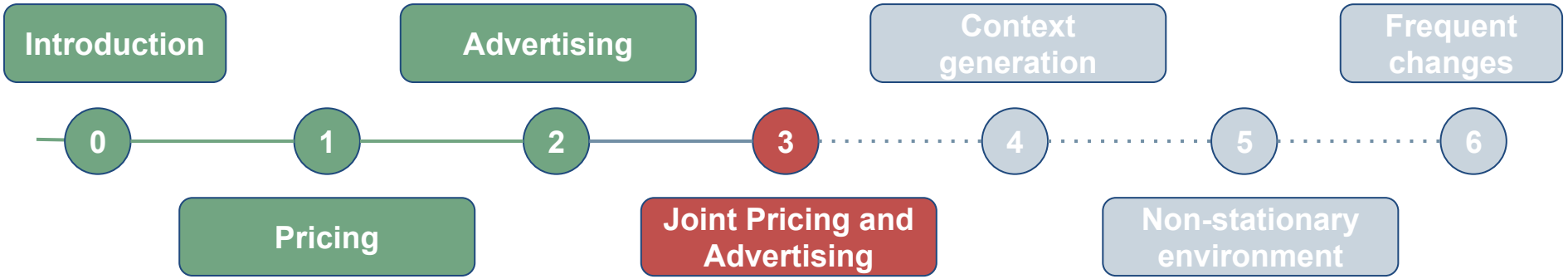








Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 3: Learning for joint pricing and advertising

Setting

Scenario:

- Users belong to **a single class** C1
- Curves related to the **advertising** problem are **unknown**
- Curve related to the **pricing** problem is **unknown**

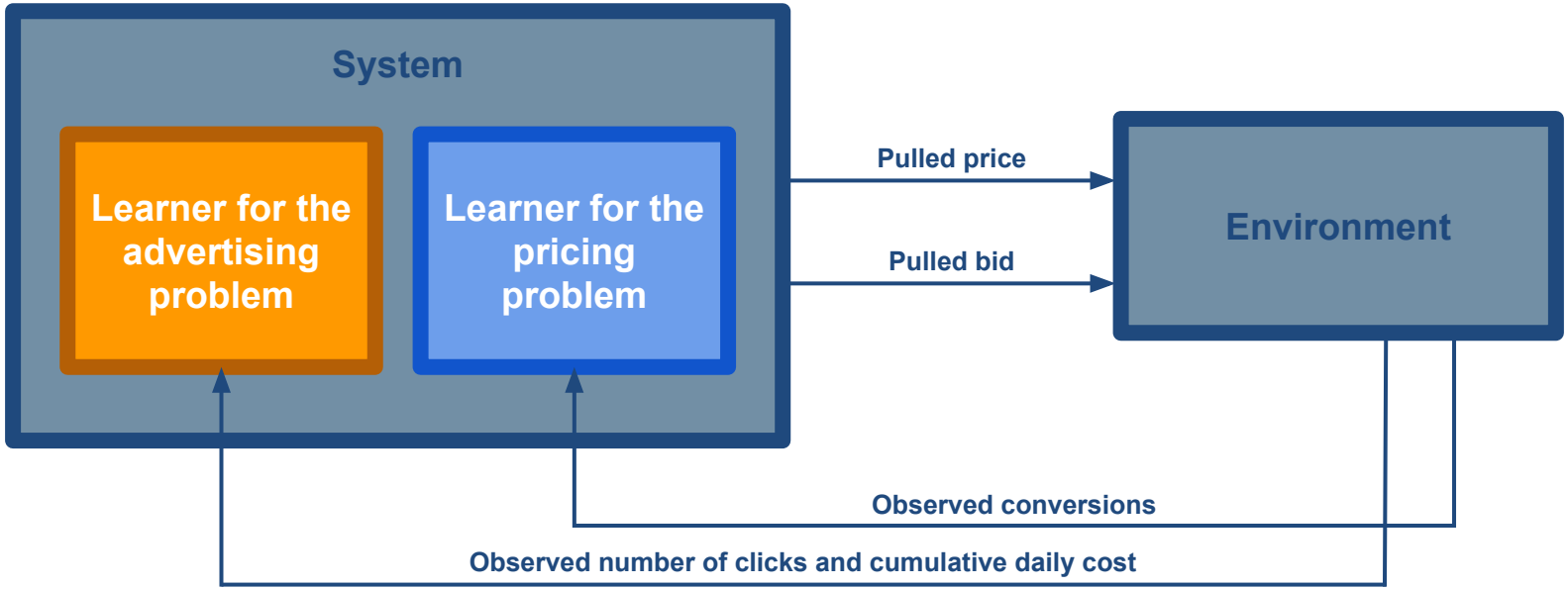
Task:

- Apply **UCB1** and **TS** to maximize the reward while estimating the conversion probabilities
- Apply **GP-UCB1** and **GP-TS** to maximize the reward while estimating the advertising curves

Step 3: Learning for joint pricing and advertising

Approach

Schema of the system



Step 3: Learning for joint pricing and advertising

Approach

The learner performs the **maximization of the reward by jointly optimizing prices and bids**.

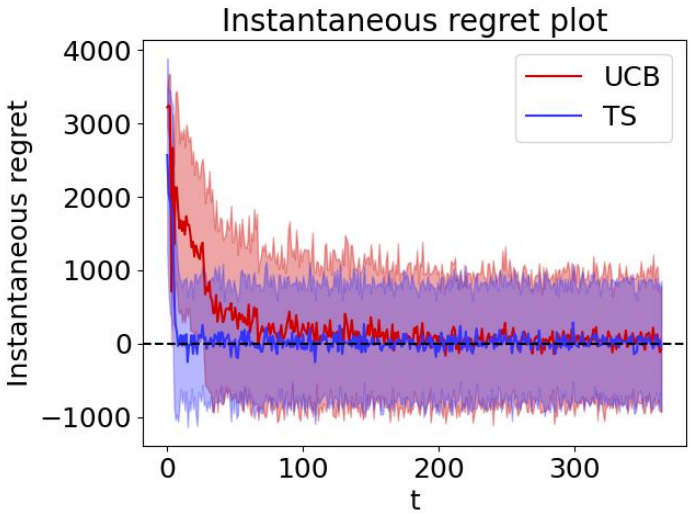
Steps of the optimization:

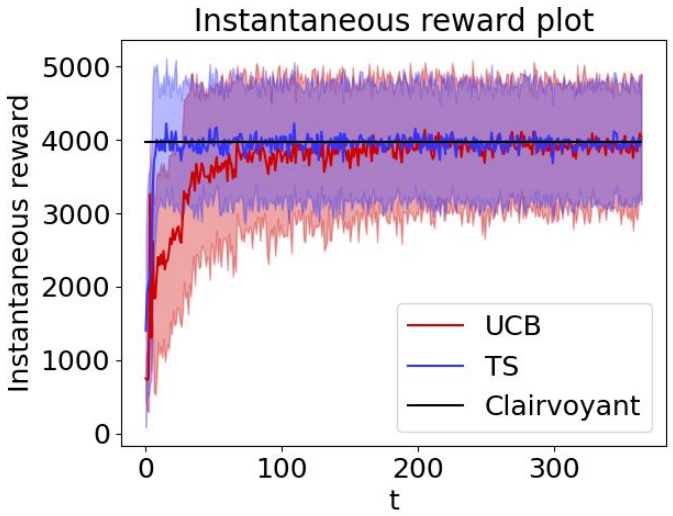
1. **compute** an estimate of the **conversion probabilities for all the prices**.
2. **compute** an estimate of the **number of clicks** and of the **cumulative daily costs for all the bids**.
3. **compute** an estimate of the **reward for all the couples price-bid** using the formula explained in the step 0;
4. the **pulled price** and **bid are the ones that maximize the reward**. Then they are played in the environment.

$$(price_{pulled}, bid_{pulled}) \leftarrow \arg \max_{(price, bid)} \{\#clicks_{est} \cdot conversion\ probability_{est} \cdot (price - other\ costs) - daily\ costs_{est}\}$$

Step 3: Learning for joint pricing and advertising

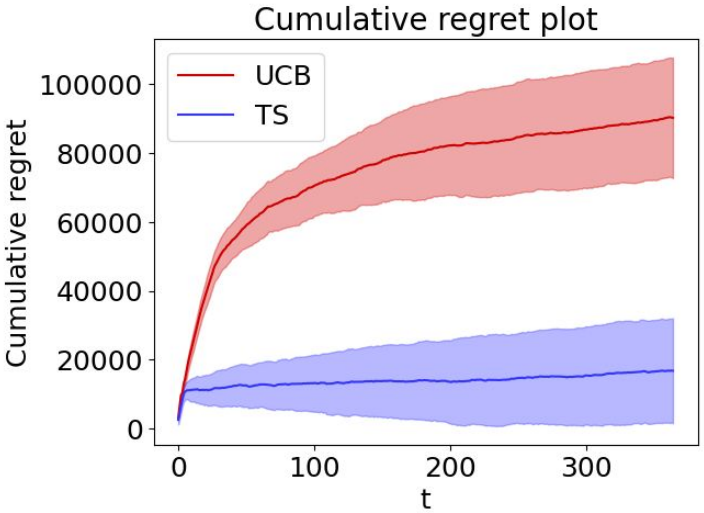
Results - Instantaneous regret and instantaneous reward

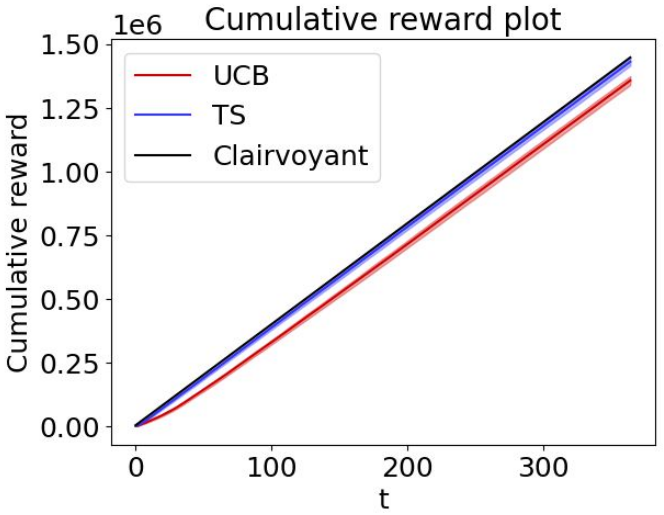




Step 3: Learning for joint pricing and advertising

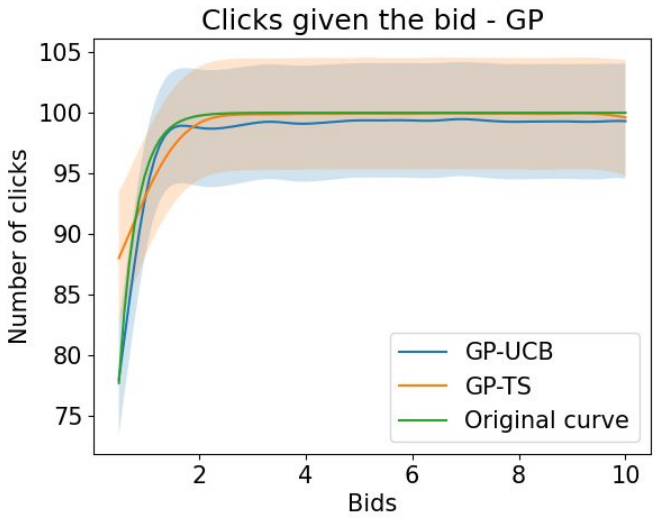
Results - Cumulative regret and cumulative reward

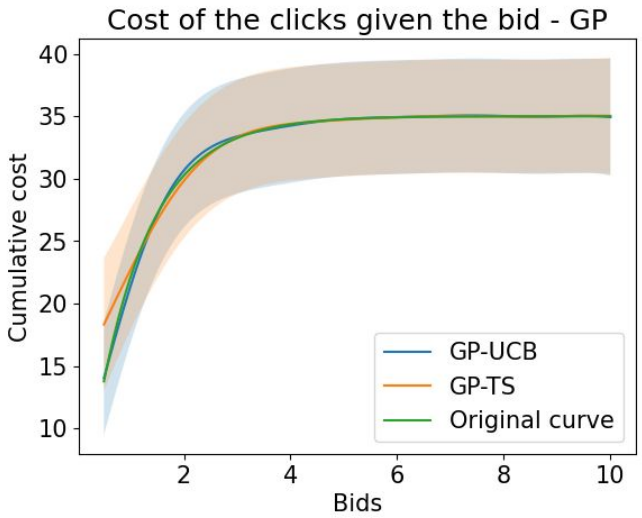




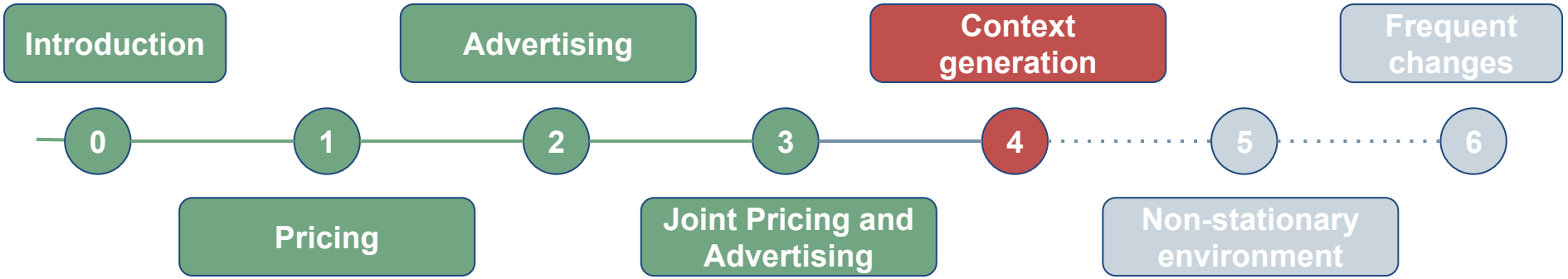
Step 3: Learning for joint pricing and advertising

Results - Estimated curves of the advertising



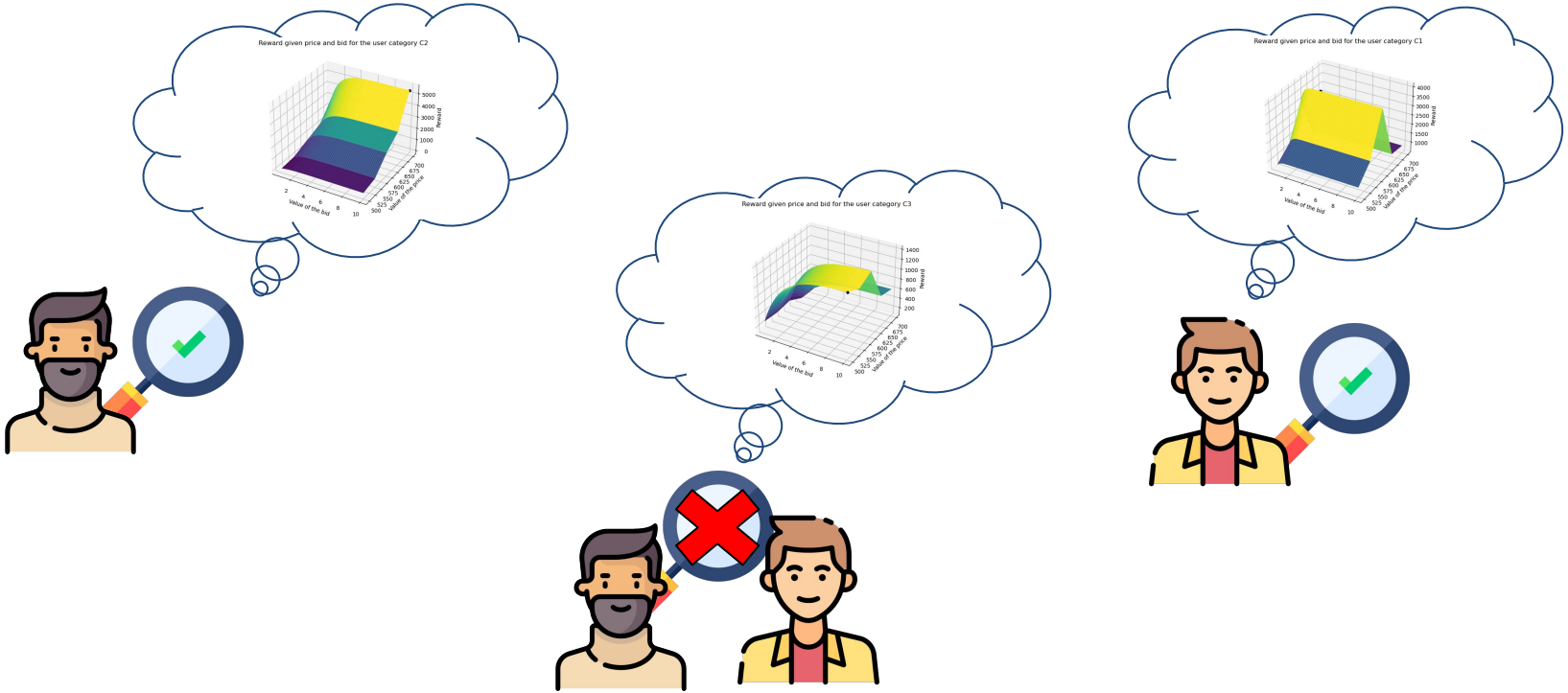


Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 4: Context generation

Setting



Step 4: Context generation

Setting - Case 1

Scenario:

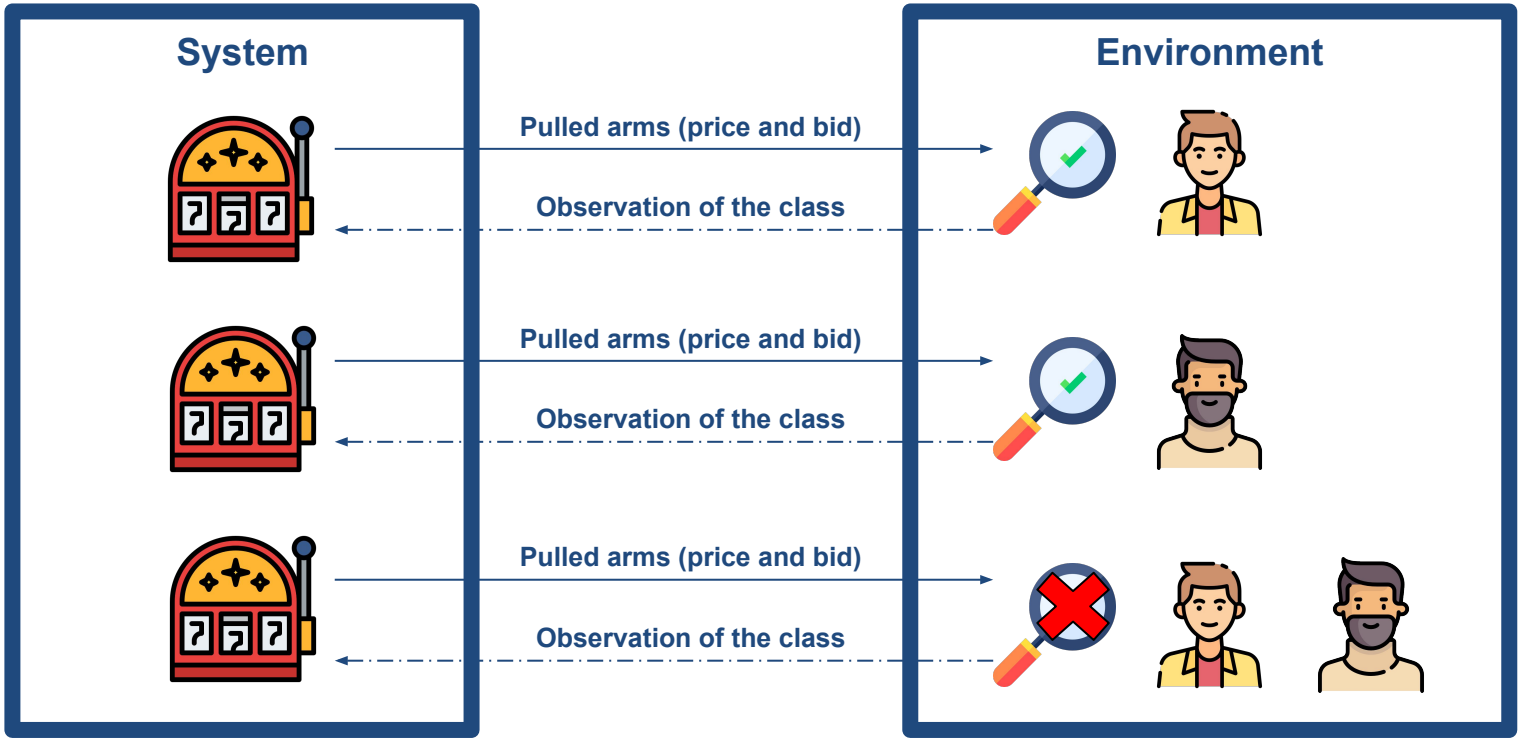
- Users belong to **three classes**: C1, C2, C3
- Users are characterized in terms of **2 binary features**
- The **context structure** is **known** beforehand
- Curves related to the **advertising** problem are **unknown**
- Curve related to the **pricing** problem is **unknown**

Task:

- Apply **UCB1** and **TS** to maximize the reward while estimating the conversion probabilities for each class separately
- Apply **GP-UCB1** and **GP-TS** to maximize the reward while estimating the conversion probabilities for each class separately

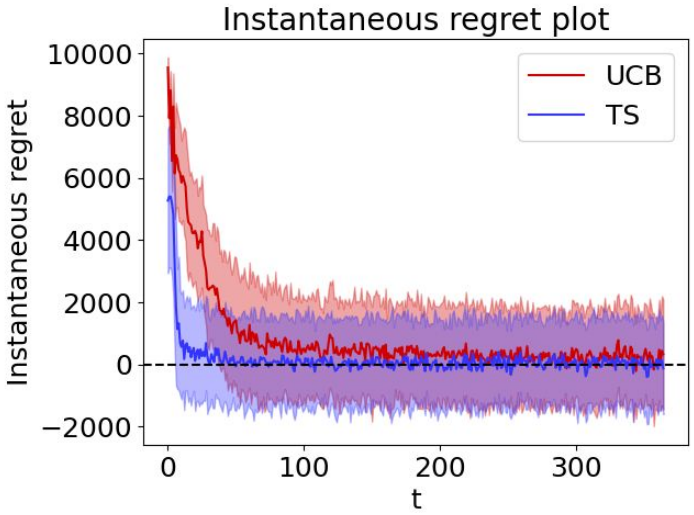
Step 4: Context generation

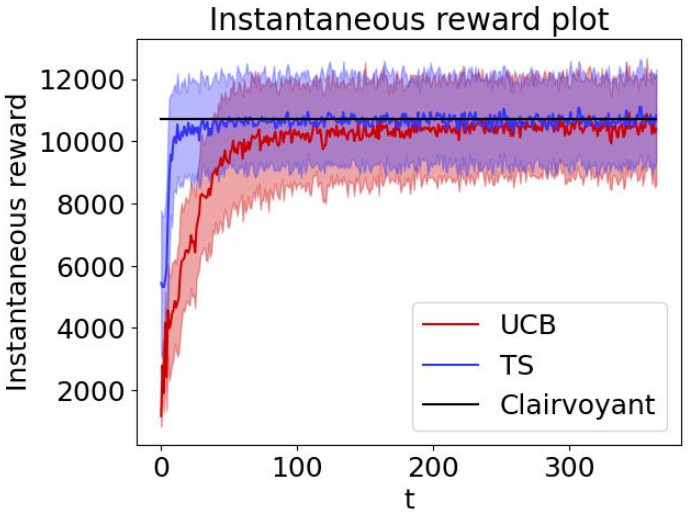
Approach - Case 1



Step 4: Context generation

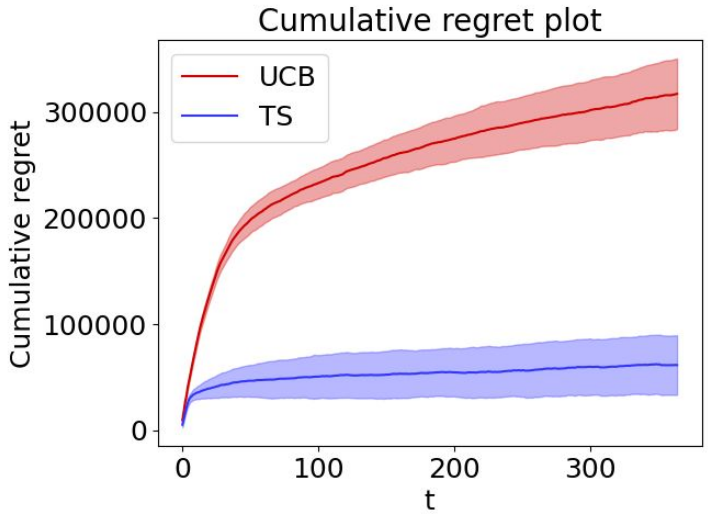
Results - Case 1 - Instantaneous regret and instantaneous reward

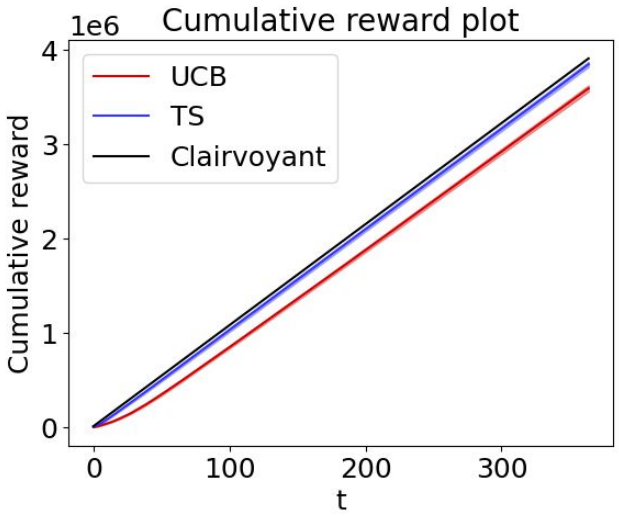




Step 4: Context generation

Results - Case 1 - Cumulative regret and cumulative reward





Step 4: Context generation

Setting - Case 2

Scenario:

- Users belong to **three classes**: C1, C2, C3
- Users are characterized in terms of **2 binary features**
- The **context structure** is **unknown**
- Curves related to the **advertising** problem are **unknown**
- Curve related to the **pricing** problem is **unknown**

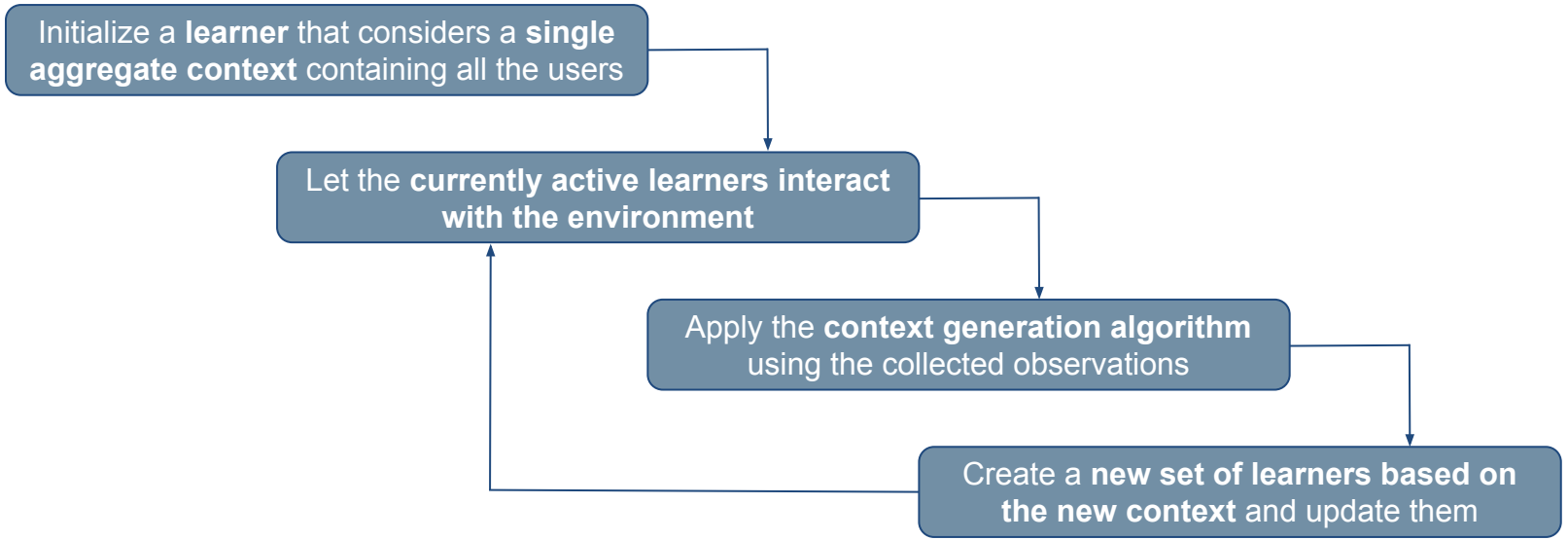
Task:

- Apply **UCB1** and **TS** to maximize the reward while estimating the conversion probabilities for each class separately
- Apply **GP-UCB1** and **GP-TS** to maximize the reward while estimating the advertising curves for each class separately
- Apply a **context generation algorithm** in order to estimate the context (every two weeks)

Step 4: Context generation

Approach - Case 2

Steps of the approach from an high level point of view

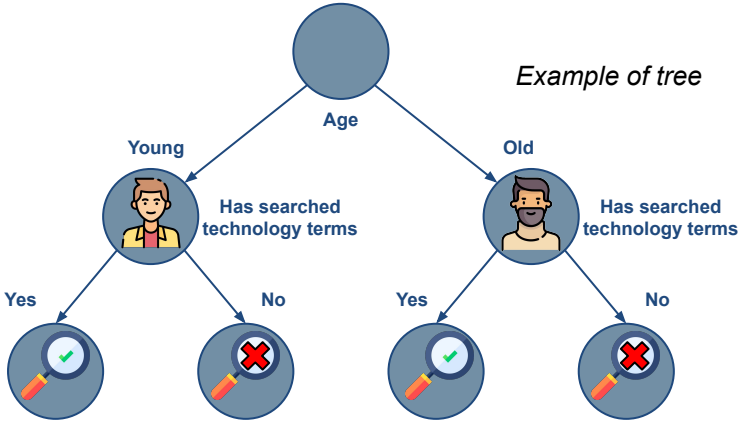


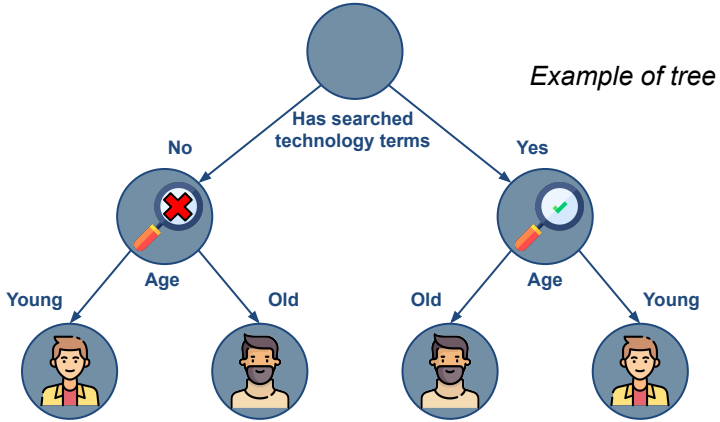
Step 4: Context generation

Greedy algorithm - Case 2

The **context generation algorithm** applied is a **greedy algorithm** that estimates the reward of a context and the sum of the rewards coming from the possible sub-context and chooses whether to split or not based on them.

The **greedy algorithm builds a tree** where **each node represent a context**. The algorithm starts from a single node, namely the **root**, that **consists in the completely aggregate context**.

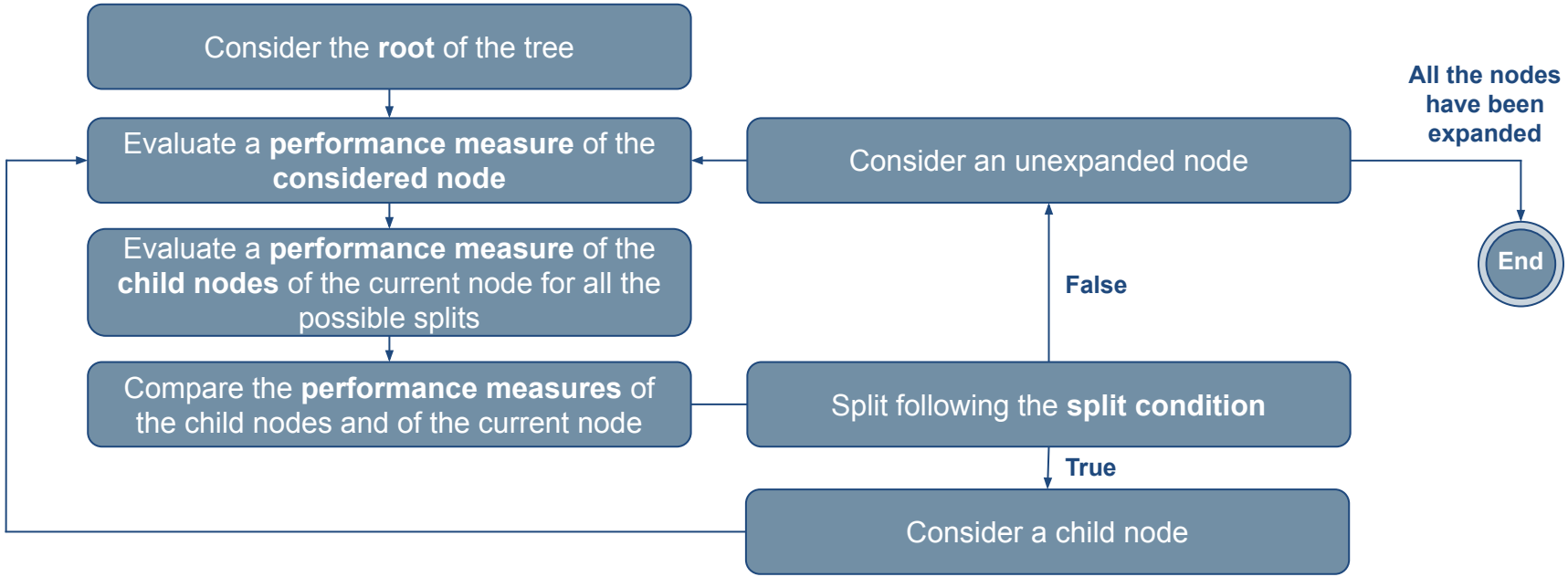




Step 4: Context generation

Greedy algorithm - Case 2

Steps of the greedy algorithm for context generation from a high level point of view



Step 4: Context generation

Split condition - Case 2

In the case of **binary features** the **split condition** that is evaluated when splitting is:

$$\underline{p}_{C_1}\underline{\mu}_{a^*_{C_1},C_1} + \underline{p}_{C_2}\underline{\mu}_{a^*_{C_2},C_2} \geq \underline{\mu}_{a^*_{C_0},C_0}$$

- $\underline{p}_{C_x}$ is the **lower bound of the probability of a user to belong to the context C_x**;
- $\underline{\mu}_{a^*_{C_x},C_x}$ is the **lower bound of the reward of the best arm in the context C_x**;
- C_1 and C_2 are two sub-contexts generated by C_0, in other words they are the child nodes of C_0 splitting on a feature.

$$\underline{reward} = \underline{\#_{clicks}} * \underline{\text{conversion probability}} * (price - \text{other costs}) - \overline{\text{cumulative daily costs}}$$

Step 4: Context generation

Lower bound - Case 2

The **lower bound** used for the **conversion probability** and for the **probability of a context** is the **Hoeffding bound**, the general formula is:

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

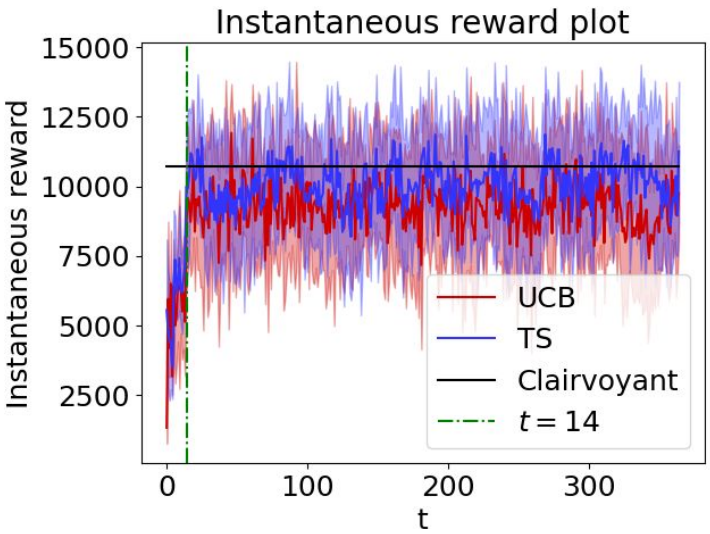
$$\underline{conversion\ probability_p} = \frac{\#convesions_{p,t}}{\#observations_{p,t}} - \sqrt{-\frac{\log \delta}{2\,\#observations_{p,t}}}$$

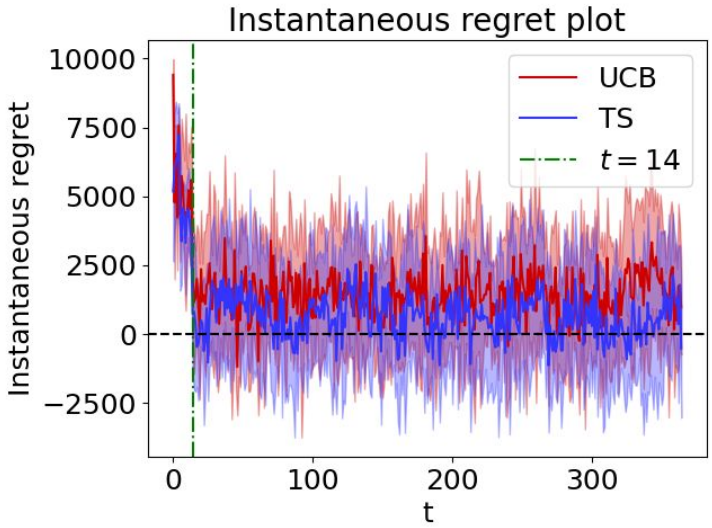
$$\underline{p}_{C_1} = \frac{\#clicks_{C_1,t}}{\#clicks_{C_0,t}} - \sqrt{-\frac{\log \delta}{2\,\#clicks_{C_1,t}}}$$

- $\bar{x}$ is the **mean** of the considered **random variable**;
- δ is 1 minus the **confidence**, it can be considered as a metric used to control the level of certainty required for an algorithm to decide whether to split the context or not;
- $|Z|$ is the **size** of the considered **set of data**.

Step 4: Context generation

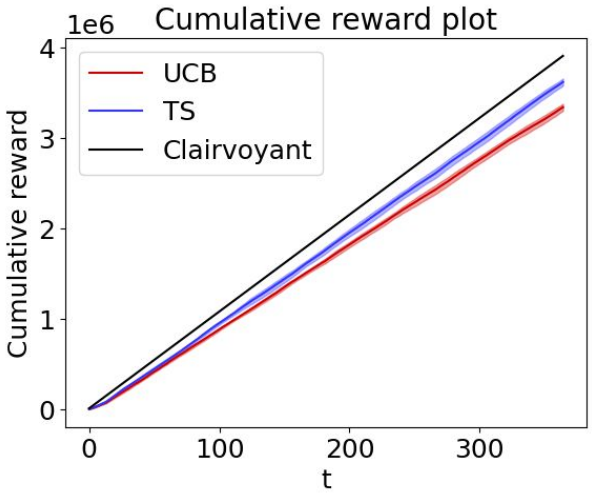
Results - Case 2

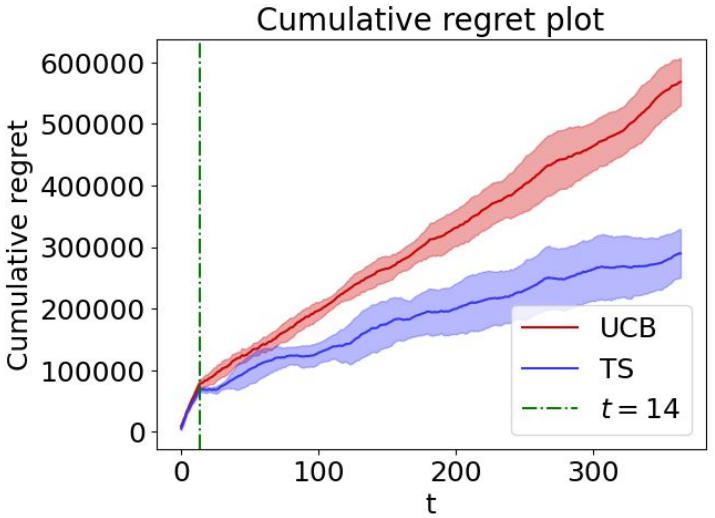




Step 4: Context generation

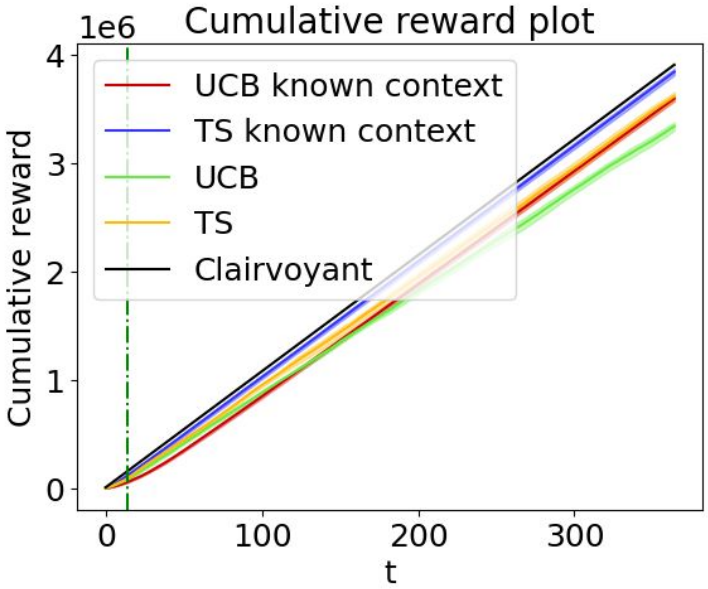
Results - Case 2

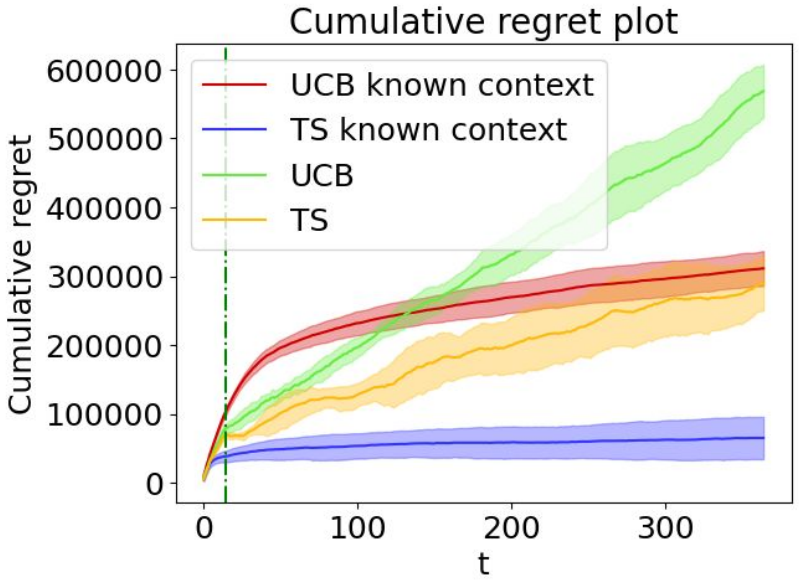




Step 4: Context generation

Comparison cases 1 and 2





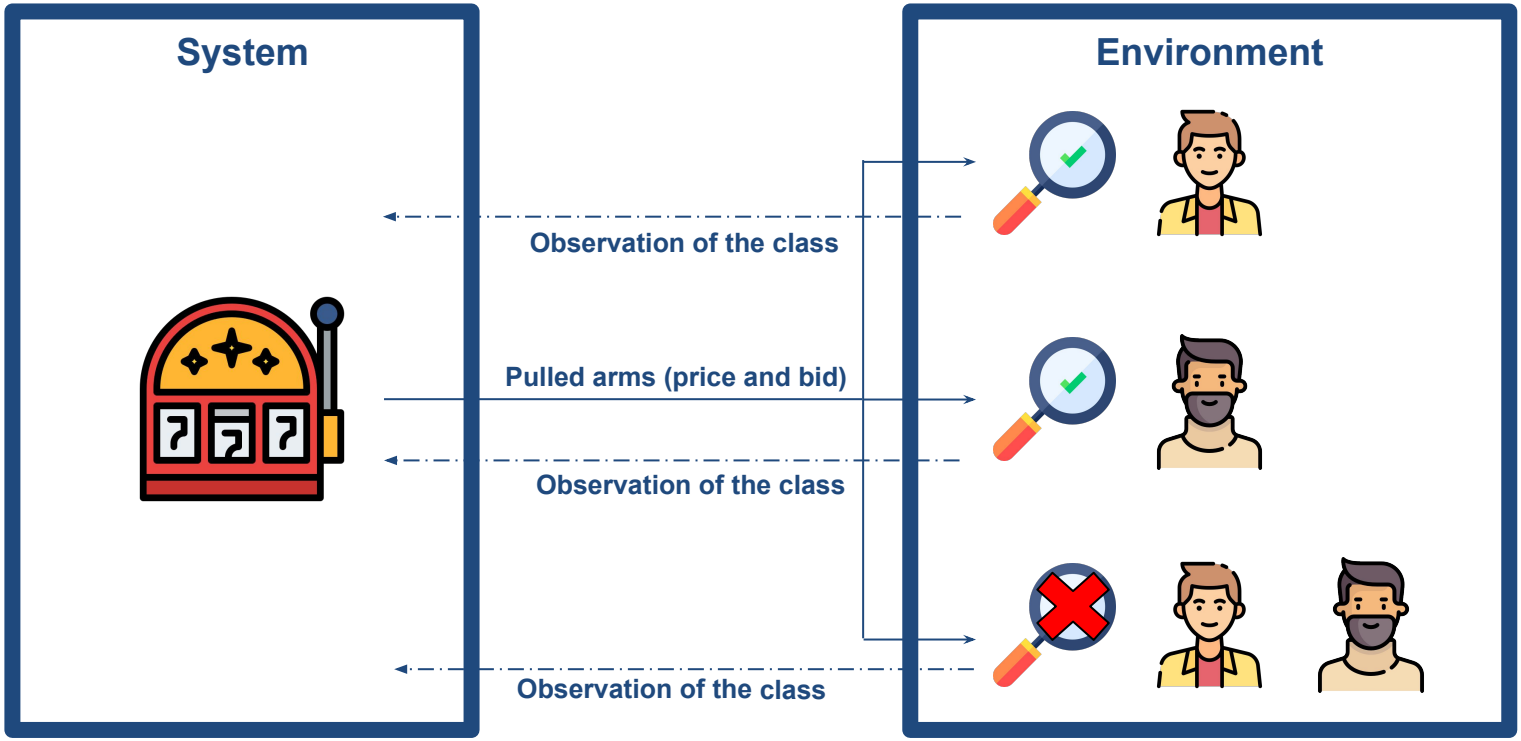
Step 4: Context generation

Setting & Approach - Case 3

- The **context structure** in this case is **unknown**, and **no context generation algorithm** is used to learn it.
- A **single learner** is employed to address the problem which only **considers** the **aggregate context**.
- The learner's objective is to **maximize** its **reward** by interacting with the environment.
- The approach is similar to step 3:
  - The learner chooses the best price and bid for maximum reward.
  - It plays the chosen price and bid in the environment.
  - The learner updates its pricing and advertising models using the obtained conversions, clicks, and cumulative daily cost.

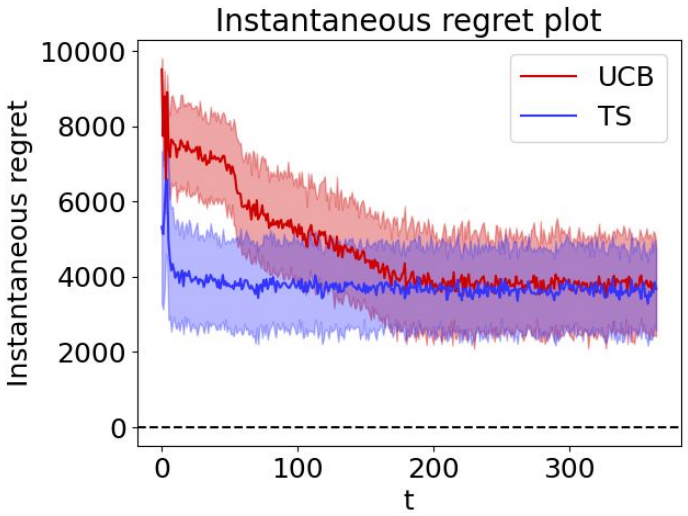
Step 4: Context generation

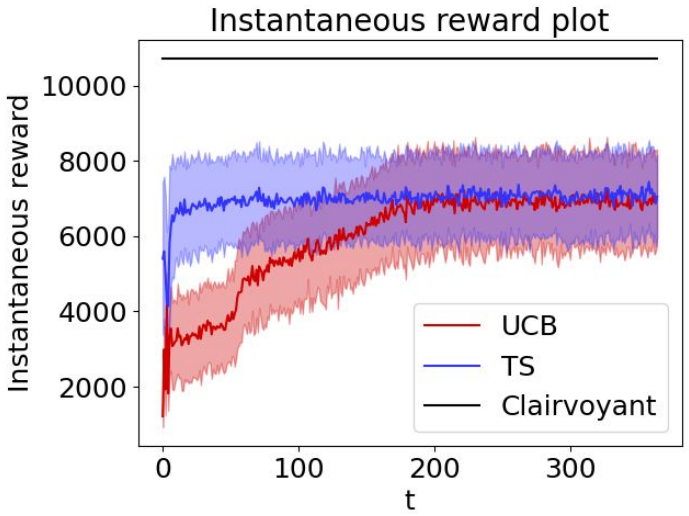
Approach - Case 3



Step 4: Context generation

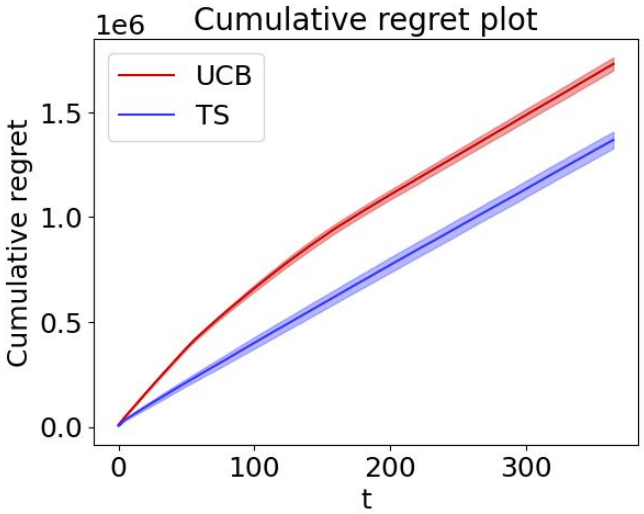
Results - Case 3 - Instantaneous regret and instantaneous reward

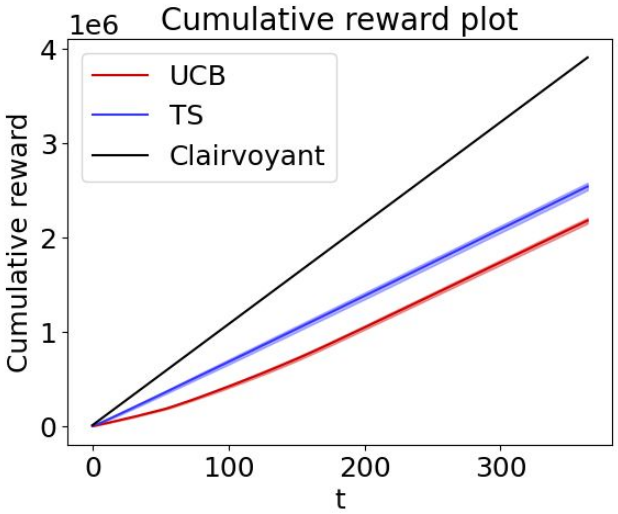




Step 4: Context generation

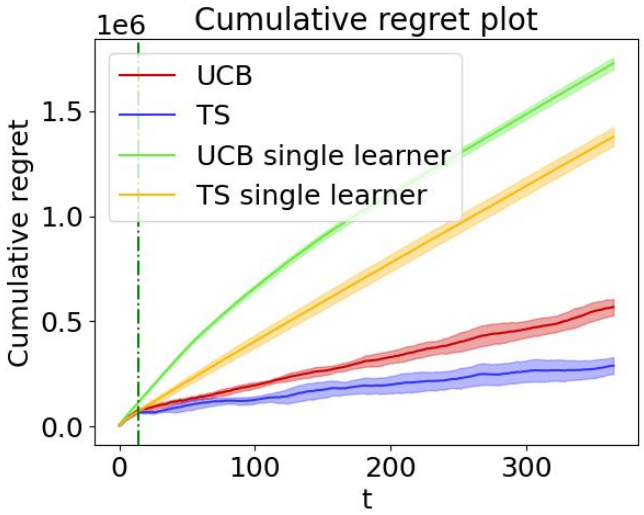
Results - Case 3 - Cumulative regret and cumulative reward

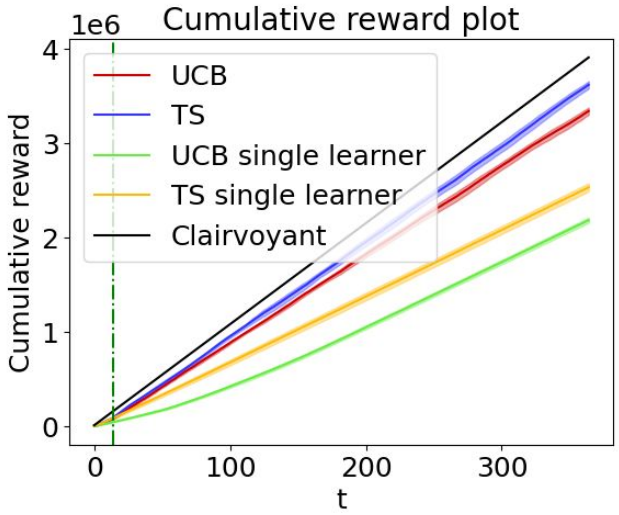




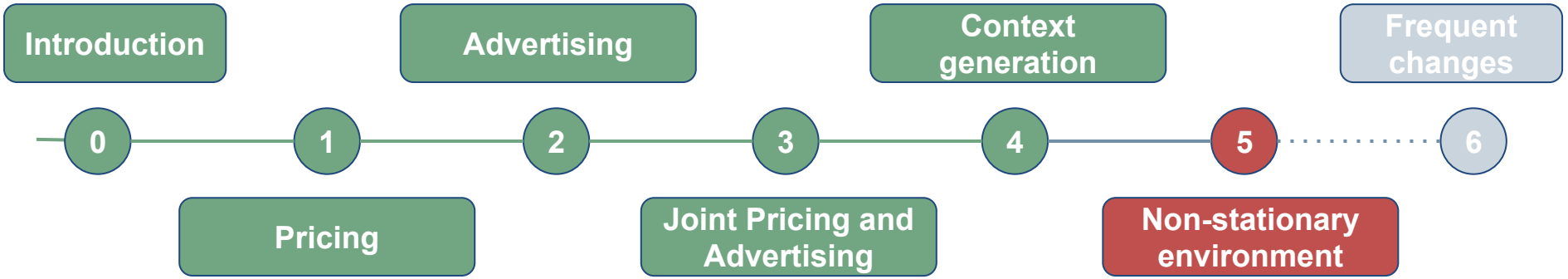
Step 4: Context generation

Comparison cases 2 and 3





Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 5: Non-stationary environments with two abrupt changes

Setting

Scenario:

- All the users belong to class **C1**
- Curves related to the advertising part are **known**
- Curves related to the pricing part are **unknown** and **non-stationary**, being subject to **three different seasonal phases**
- Apply **UCB1**, **UCB1 with sliding window** and **UCB1 with change detection test**

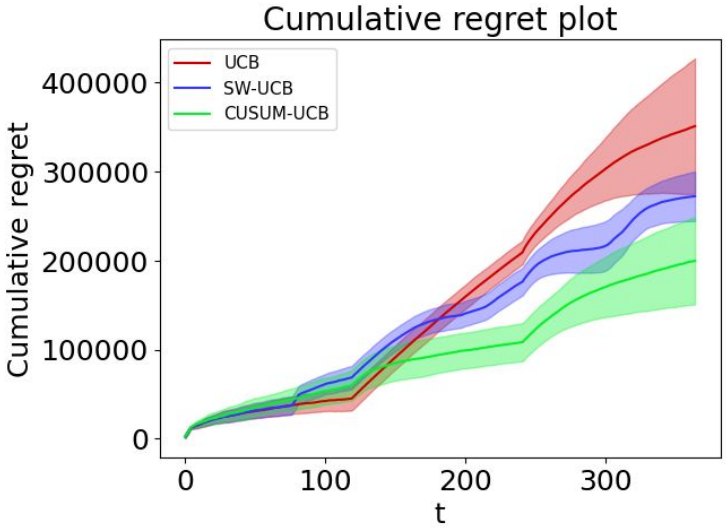
The **phases** are spread over the year as follows:

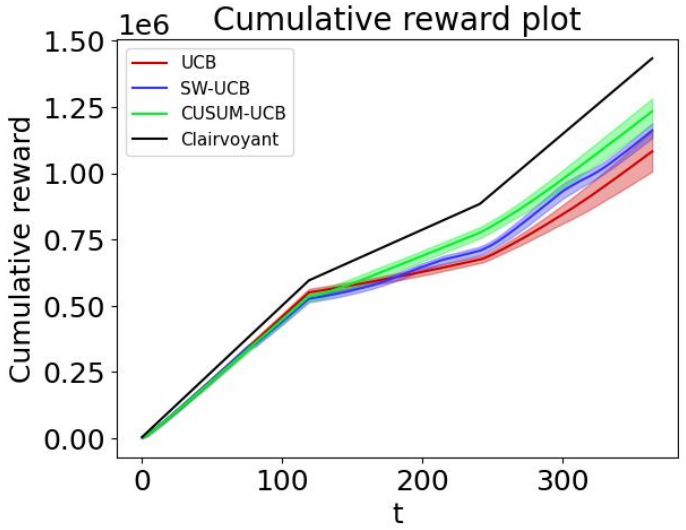
1. **first phase**, February - May, **launch of the mobile phone,** preferred price is 650€
2. **second phase**, June - September, **regular purchasing period**, preferred price is 600€
3. **third phase**, October - January, **discounts sales**, preferred price 550€

[A3] Conversion probabilities for the three phases

Step 5: Non-stationary environments with two abrupt changes

Results

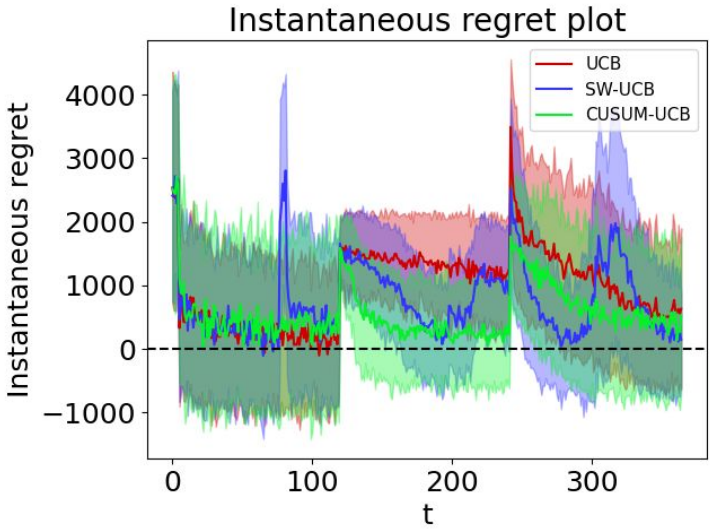


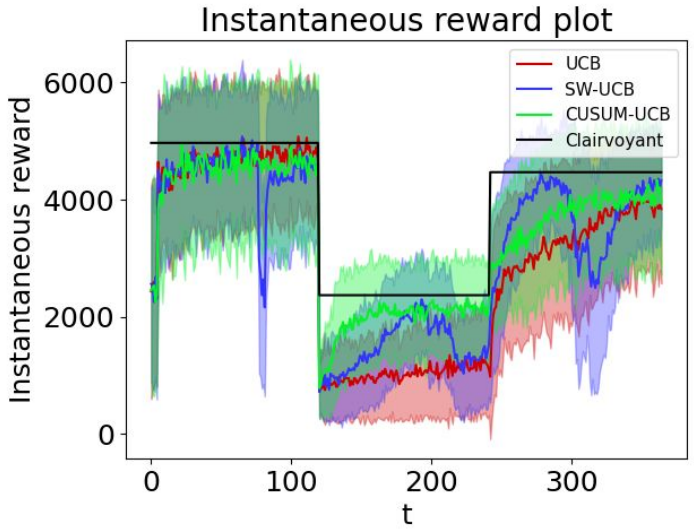


- **Two non-stationary versions** perform **better** than the vanilla UCB
- **SW-UCB** and **CUSUM-UCB** obtain a **sub-linear regret** in each phase

Step 5: Non-stationary environments with two abrupt changes

Results





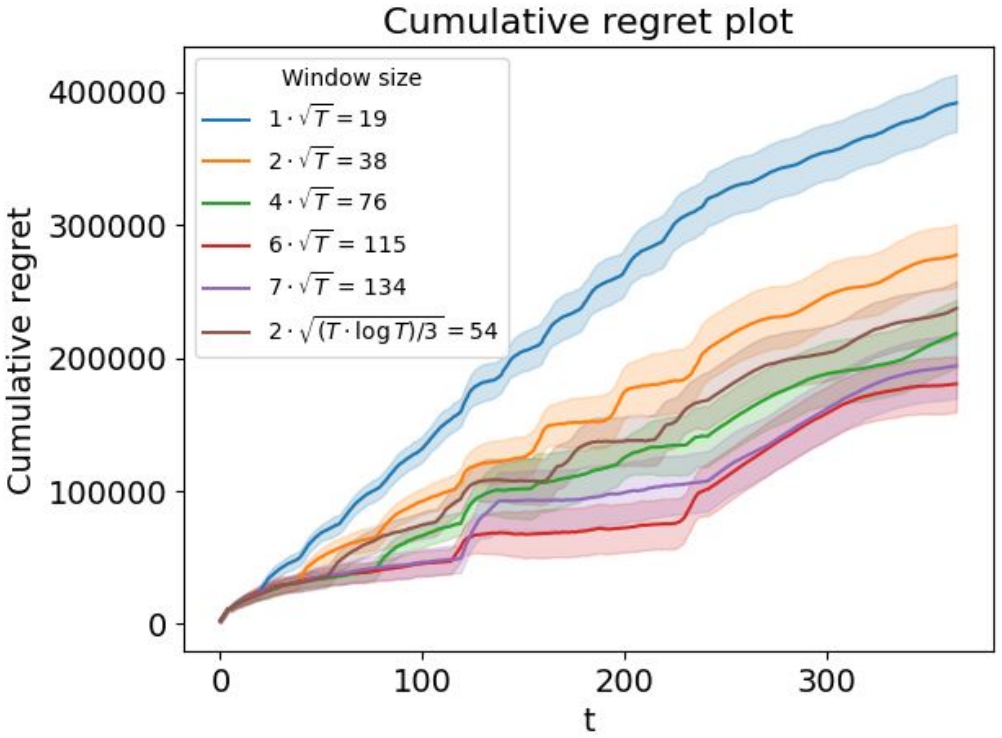
- After 120 days UCB bounds are tight so it takes a lot of time to realise that the optimal price has changed

- Periodically, SW-UCB has negative **peaks** in the instantaneous reward because **suboptimal arms leave the window** therefore the **algorithm is forced to play them at least once**

Step 5: Non-stationary environments with two abrupt changes

Sensitivity Analysis - UCB-SW

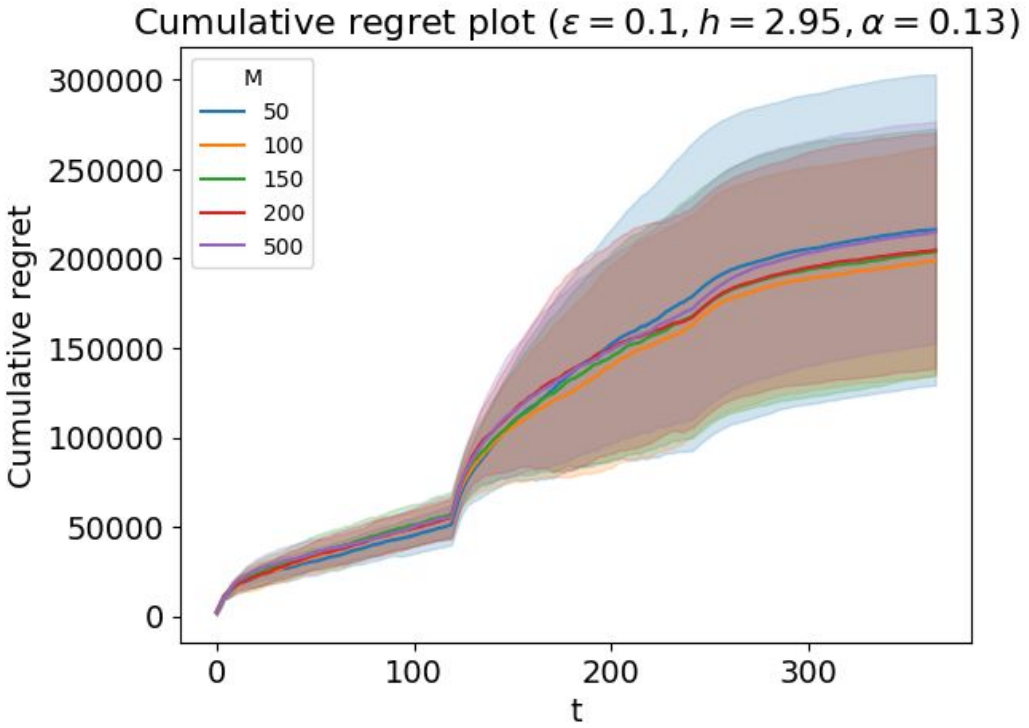
- With **small multiplicative factors low performances** because the window is too short with respect to the length of the phases
- Values for the window in the range between 50 and 80 are good.
- Using as multiplicative factor 6 or 7 would be **optimal** because the window size would be similar to the length of the phases
- With **higher factors** SW-UCB would not explore enough leading to a decrease in the performance



Step 5: Non-stationary environments with two abrupt changes

Sensitivity Analysis - UCB-CUSUM

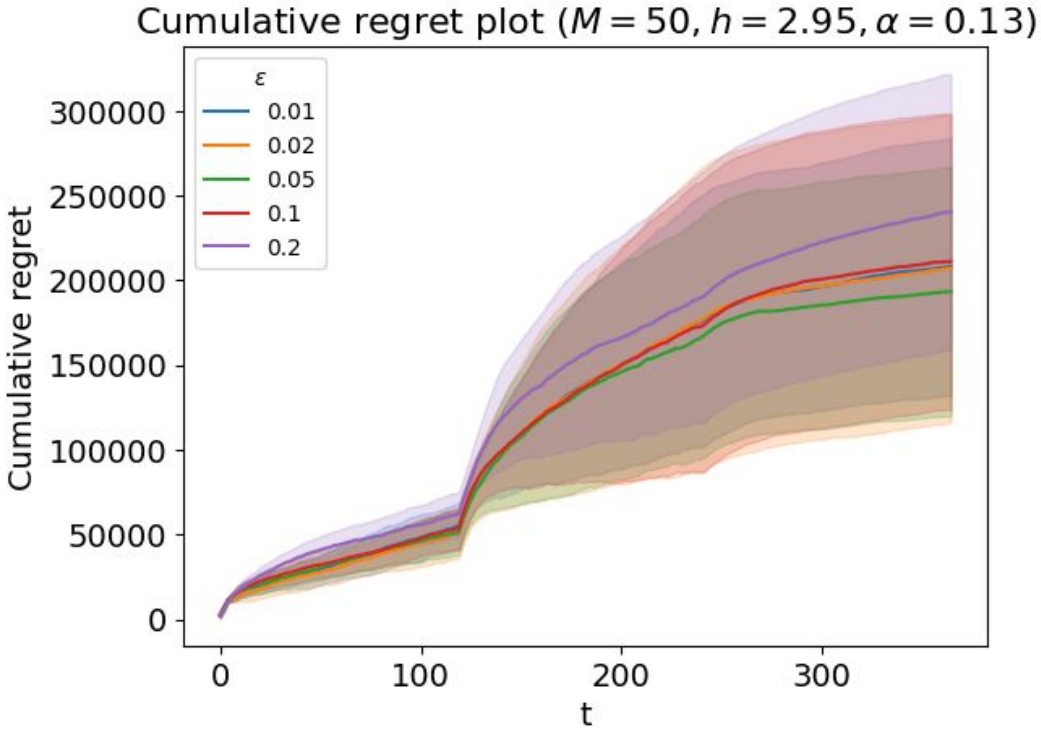
- With large **M** the **majority of the observations** are **used** to **compute** the **reference point**. So it takes a long time to identify changes
- A **too small M** leads to **inaccurate estimations of the reference point** and **more frequent change detections**
- Placing M between 50 and 100 is sufficient to get a decent estimate of the reference point
- All of the values taken into consideration perform similarly



Step 5: Non-stationary environments with two abrupt changes

Sensitivity Analysis - UCB-CUSUM

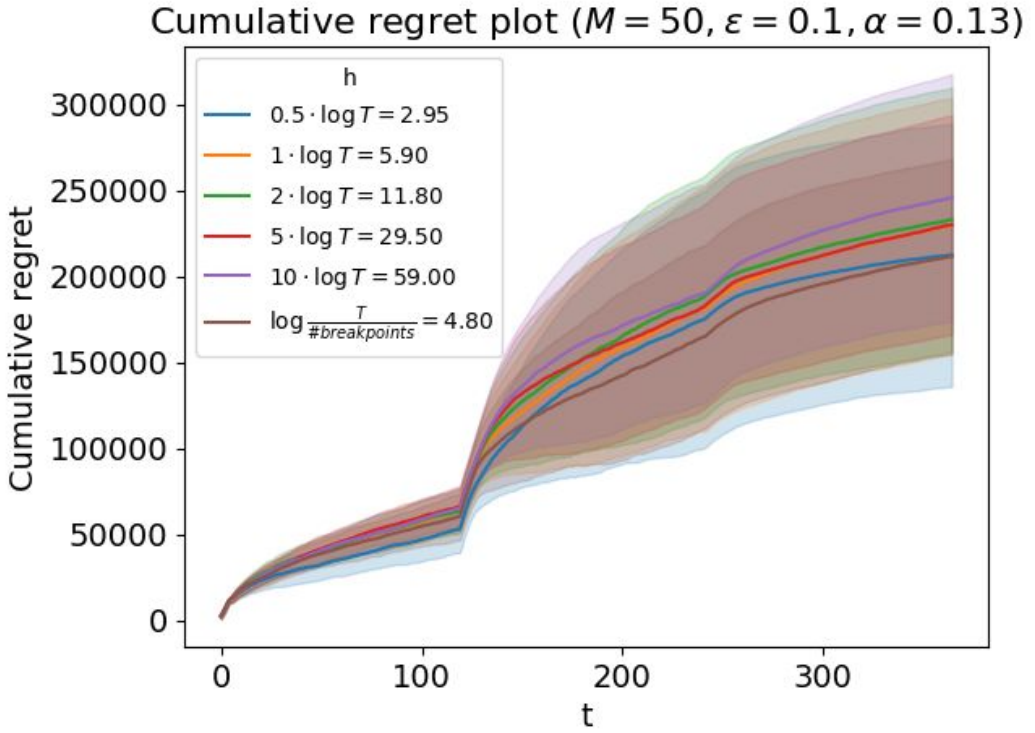
- With **low values of ε, changes are detected even when there aren't**
- As ε **increases** the **sensitivity decreases**, so more samples are necessary to flag a change.
- ε should be of the **order of the changes in the distributions**, which in our case are of the order of 0.1
- Values in the range between 0.01 and 0.1 produce similar performance



Step 5: Non-stationary environments with two abrupt changes

Sensitivity Analysis - UCB-CUSUM

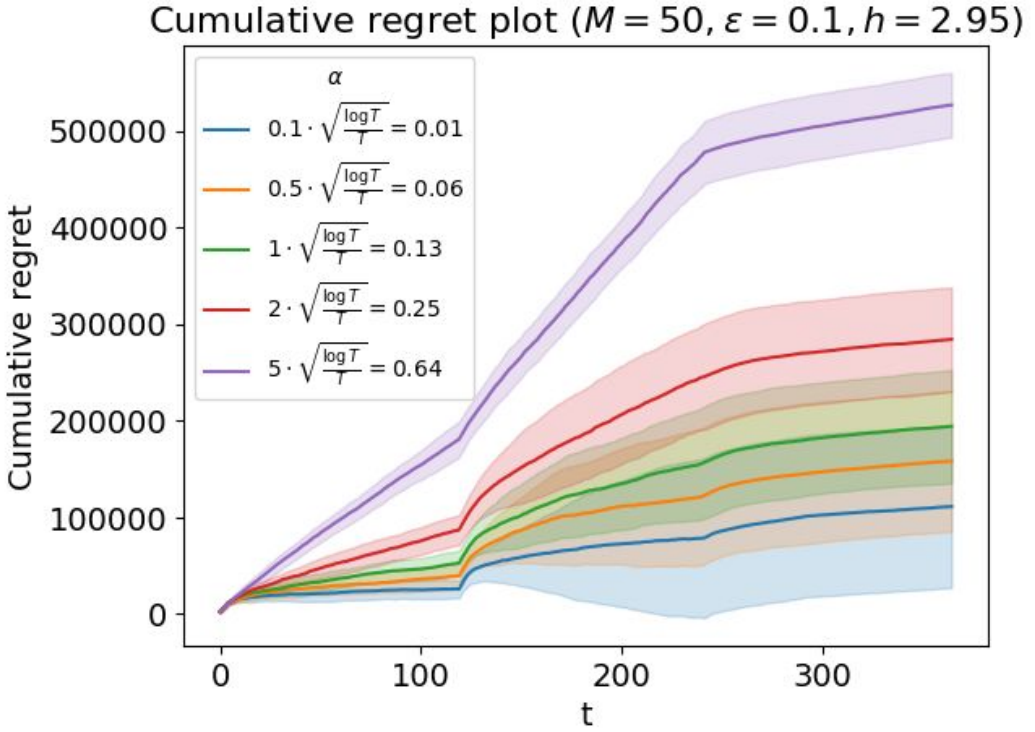
- **Low values of h** makes the **algorithm more sensitive**
- As the **factor increases** the **cumulative regret gets worse** since the learner needs more evidence to detect a change in the distribution
- **Best performing h** is the one with **multiplicative factor 0.5**



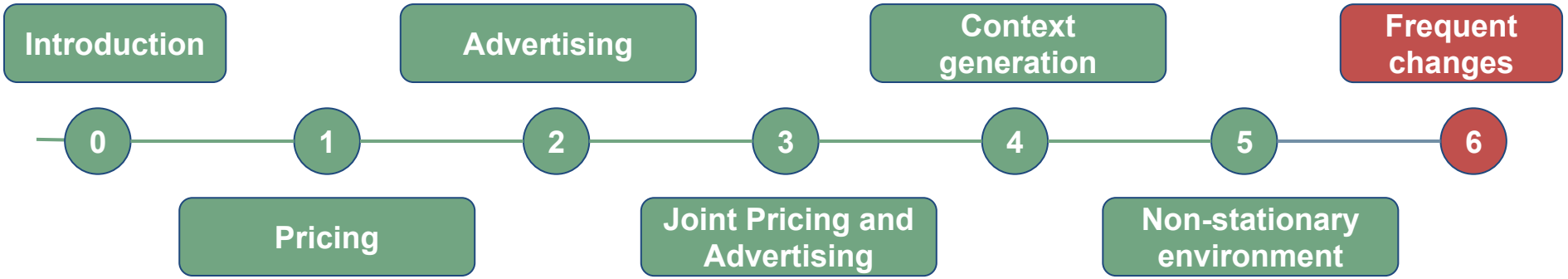
Step 5: Non-stationary environments with two abrupt changes

Sensitivity Analysis - UCB-CUSUM

- If α is **too low**, the **algorithm exploits too much**, pulling almost always the suggested arm
- If α is **too large**, the **learner explores too much** being able to detect changes, but **not exploiting what has learned**
- Lower values of α give lower regret



Roadmap


Introduction
Advertising
Context generation
Frequent changes
0
1
2
3
4
5
6
Pricing
Joint Pricing and Advertising
Non-stationary environment

Step 6: Non-stationary environments with many abrupt changes

Setting 1

Scenario:

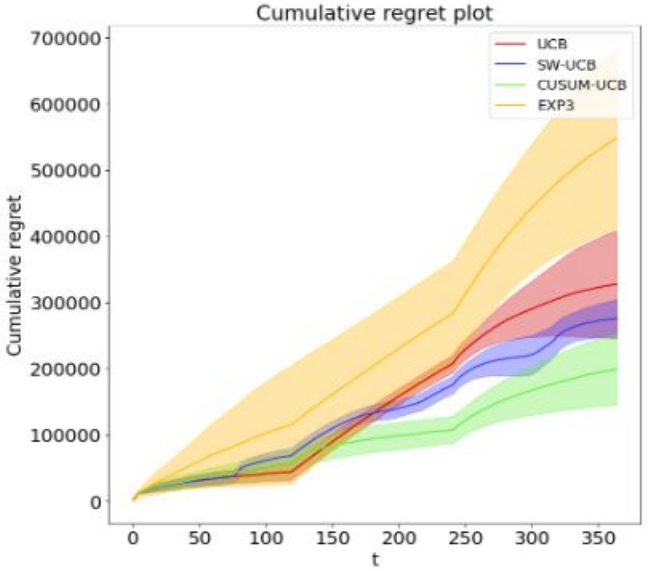
- All the users belong to class **C1**
- Advertising curves are **known**, the **bid** is **fixed**
- Curves related to the pricing part are **unknown** and **non-stationary**, being subject to **three different seasonal phases**

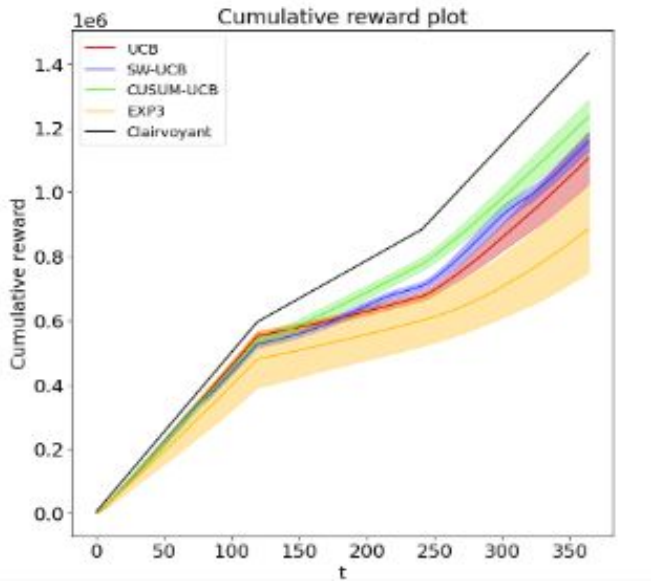
Task:

- Apply **UCB1**, **UCB1 with sliding window**, **UCB1 with change detection test** and **EXP3**

Step 6: Non-stationary environments with many abrupt changes

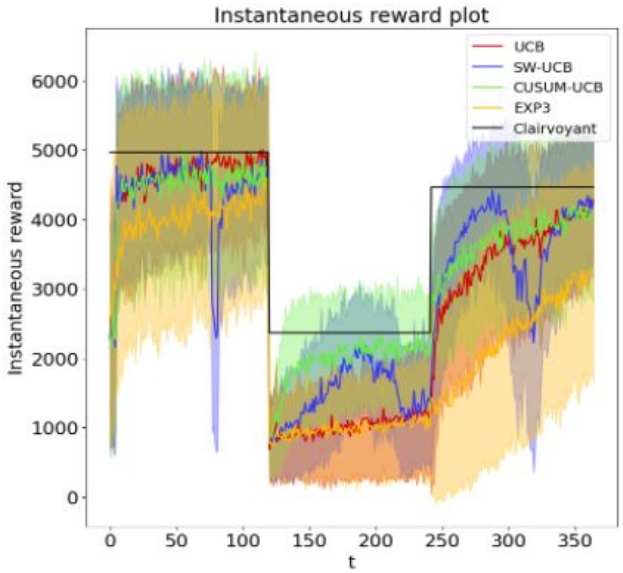
Results - Setting 1

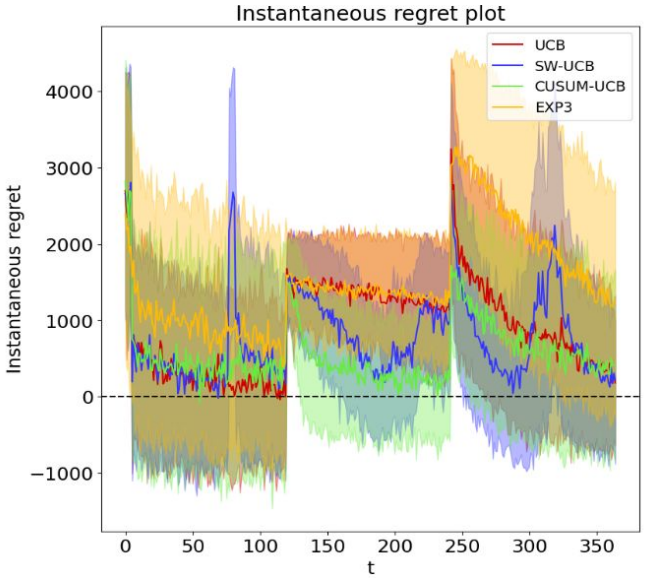




Step 6: Non-stationary environments with many abrupt changes

Results - Setting 1





Step 6: Non-stationary environments with many abrupt changes

Setting 2

Scenario:

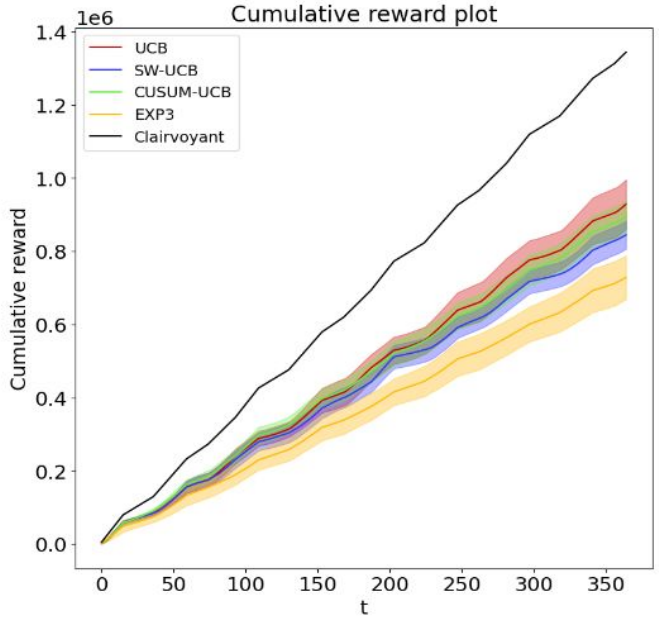
- All the users belong to class **C1**
- Advertising part are **known**, the **bid** is **fixed**
- Curves related to the pricing part are **unknown** and **non-stationary**, being subject to **five different phases** that **cyclically change** with a **high frequency**

Task:

- Apply **UCB1**, **UCB1 with sliding window**, **UCB1 with change detection test** and **EXP3**.

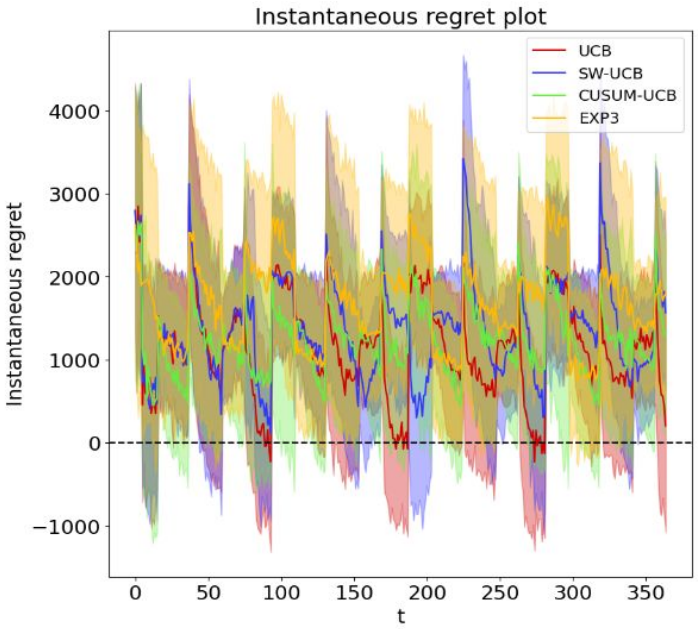
Step 6: Non-stationary environments with many abrupt changes

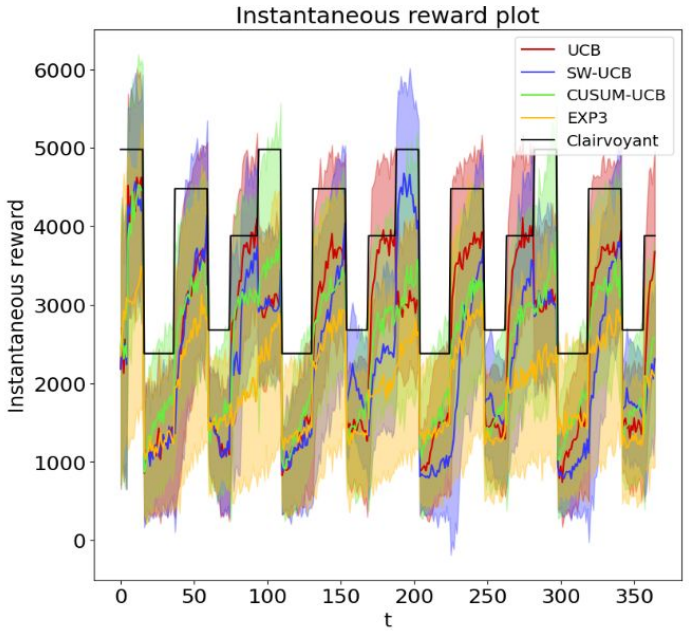
Results - setting 2



Step 6: Non-stationary environments with many abrupt changes

Results - setting 2






POLITECNICO
MILANO 1863

References

- [1] Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design, Srinivas et al. (2010), https://arxiv.org/pdf/0912.3995.pdf
- [2] On Upper-Confidence Bound Policies for Non-Stationary Bandit Problems,Garivier, A., & Moulines, E. (2008), https://arxiv.org/pdf/0805.3415.pdf
- [3] A Change-Detection based Framework for Piecewise-stationary Multi-Armed Bandit Problem, Liu et al., 2017, https://arxiv.org/pdf/1711.03539.pdf
- [4] https://jeremykun.com/2013/11/08/adversarial-bandits-and-the-exp3-algorithm/

Appendix - A1

⇦ [A1] Here it follows the precise numbers used for the conversion probabilities of the three classes for step 1 to 4:

Conversion Probabilities per class					
Class \ Price (€)	500	550	600	650	700
C1	0.10	0.12	0.20	0.04	0.03
C2	0.03	0.04	0.10	0.12	0.20
C3	0.20	0.12	0.10	0.04	0.03

Steps 5 and 6 require a detailed analysis on non-stationary settings for a single class, thus these values change. View the relevant appendices.

Appendix - A2

⇦ [A2] Here the equations used to plot the clicks and costs curves:

$$f(bid) = scale * (1 - e^{-slope*bid}),\ bid \in [0.5, 10]$$

Clicks		
Class \ Parameters	Scale	Slope
C1	100	3
C2	90	1
C3	70	0.5

Cumulative daily costs		
Class \ Parameters	Scale	Slope
C1	35	1
C2	25	0.5
C3	15	0.2

Appendix - A3

⇦ [A3] Here it follows the precise numbers used for the conversion probabilities of the three phases for step 5 and step 6 case 1:

Conversion Probabilities per phase					
Phase \ Price (€)	500	550	600	650	700
Phase1	0.07	0.10	0.10	0.20	0.10
Phase2	0.08	0.06	0.12	0.03	0.03
Phase3	0.18	0.30	0.15	0.02	0.02

Appendix - A4

⇦ [A4] Here it follows the precise numbers used for the conversion probabilities of the five phases for step 6 case 2:

Conversion Probabilities per phase					
Phase \ Price (€)	500	550	600	650	700
Phase1	0.07	0.10	0.10	0.20	0.10
Phase2	0.08	0.06	0.12	0.03	0.03
Phase3	0.18	0.30	0.15	0.02	0.02
Phase4	0.27	0.12	0.06	0.02	0.01
Phase5	0.09	0.06	0.12	0.10	0.13