Toward reducing cumulative bias in automated decision-making systems using Multi-Armed Bandits

TREBALL FI DE GRAU DE

Quim De Las Heras Molins

Director: Javier Segovia Aguas

Grau en Enginyeria en Informàtica

Curs 2021-2022

upf. Universitat Pompeu Fabra Barcelona

Escola d'Enginyeria

Acknowledgements

I would like to express my many thanks to my supervisor, Javier Segovia Aguas, for their predisposition to aiding me with this project and their support and guidance all throughout its materialization.

I am also very appreciative of all the people, fellow students and professors, I met during the year-long stay in Maynooth University during which I worked on this and who offered me help whenever I needed it.

Lastly, I would like to thank my family and friends, for their endless care and advice, and for being there as role-models to me the whole time I was pursuing this degree.

Abstract

This project aims to showcase the harmful cumulative effect Artificial Intelligence (AI) bias can have on automated decision-making systems and their non-stationary environments, and propose some methods toward addressing it. Whilst using a Multi-Armed Bandit algorithm as our model to represent one instance of said automated decision-making systems, we present four original constraints to mitigate the discrimination manifest in our case of study. That is, basing our efforts on a simulated environment based on some key features from a selection of real data from the US mortgage lending HMDA database, we examine how each constraint performs within an updated version of the UCB algorithm in succeeding to reduce the rise of said cumulative bias effect, and even amending the initial bias present within the input.

Resum

Aquest projecte té com a objectiu exposar les conseqüències acumulatives que els biaixos en Intel·ligència Artificial (IA) poden provocar sobre els sistemes automatitzats de presa de decisions i els seus entorns no estacionaris, i proposar alguns mètodes per abordar aquest problema. Utilitzant un algoritme Multi-Armed Bandit com a model per representar un d'aquests sistemes automatitzats de presa de decisions, es presenten quatre restriccions originals per mitigar la discriminació existent en el nostre cas d'estudi. És a dir, basant els nostres esforços en un entorn simulat basat en algunes característiques clau d'una selecció de dades reals de la base de dades HMDA de préstecs hipotecaris als EUA, s'analitzarà com cadascuna d'aquestes restriccions es comporta quan s'utilitza com a part d'una versió actualitzada de l'algoritme UCB per aconseguir reduir aquest biaix acumulatiu, i fins i tot corregir el biaix inicial present en les dades.

Resumen

Este proyecto tiene como objetivo demostrar las consecuencias acumulativas que los sesgos en Inteligencia Artificial (IA) pueden provocar sobre los sistemas automatizados de toma de decisiones y sus entornos no estacionarios, y proponer algunos métodos para abordar este problema. Utilizando un algoritmo Multi-Armed Bandit como modelo para representar uno de estos sistemas automatizados de toma de decisiones, se presentan cuatro restricciones originales para mitigar la discriminación existente en nuestro caso de estudio. Es decir, basando nuestros esfuerzos en un entorno simulado basado en algunas características clave de una selección de datos reales de la base de datos HMDA de préstamos hipotecarios en EE. UU, se analizará cómo cada una de estas restricciones se comporta cuando se utiliza como parte de una versión actualizada del algoritmo UCB para conseguir reducir este sesgo acumulativo, e incluso corregir el sesgo inicial presente en los datos.

Contents

1. INTRODUCTION

As Artificial Intelligence (AI) is being rapidly adopted across an increasing variety of sectors including health care, the labor market and justice, the need to make sure such algorithms and technologies being deployed are safe and fair to use is similarly ever growing.

The automation of these decision-making processes has brought with it legitimate ethical concerns not only about the discrimination they are able to enforce when being based on unacceptable sense-making models (Angwin et al., 2016; Dastin, 2018; Ledford, 2019), but also the reduced feeling of accountability behind these ill-advised decisions and their repercussions when they take place (Hevelke and Nida-Rümelin, 2015). That is, an increase in the vulnerability those mistreated by the algorithms might feel when wrongfully dealt with by a faulty automated system.

Indeed, unfair decisions and prejudiced systems can also be identified within the real-world and in human-only domains, as they have been in place for way longer and earlier than AI-based ones. However, while it can be argued that there have been progresses made in almost all contexts and axes of historical injustice over the years, the same attentive scrutiny has to now be put in the AI field in order not to lose ground in any of those same fronts, or even stop pushing for more fair, equitable and unbiased processes just because of an assumption of impartiality is given to these new "disinterested" agents.

In fact, bias detected in AI systems can come from many sources along its entire design pipeline, and be of several types depending on at which stage they are being introduced (Srinivasan and Chander, 2021). In doing so, bias in AIs holding positions of power, as any decision-making role does, can still cause real harm to individual people (O'Neill, 2017), as well as have overarching effects on the environment around them. It is in this sense that particular attention must be paid to how these automated decisions, applied detached from sentiments and in mass, can have cumulative consequences that may affect the future for a given group or groups of real people in more permanent and profound ways than expected.

As any problem, and therefore their solution, is susceptible to be solved inadequately possibly due to biased premises, special care must be taken to make sure that those minorities within the data, more in risk of being affected more harshly by the worst outcomes, are protected from the perpetuation, or escalation, of said biases. An intersectional perspective, then, can be useful when keeping in mind all the interconnected ways people can be disfavored by their environment, and when ensuring that group fairness is being addressed properly.

We would certainly hope to reduce algorithmic bias as much as possible, or keep it from happening altogether. But this is, indeed, a difficult thing to accomplish for a number of reasons. One of the reasons frequently brought up for this has to do with how Machine Learning (ML) algorithms need great amounts of data to be trained, and how most data sources large enough to feed them like so will precisely be the ones that will have taken the least measures to quality-check their data for any bias or discrimintation present within (Baeza-Yates, 2016). A different array of reasons for this difficulty has to do with how the agents that develop or deploy these algorithms do not always have the public's best interest in mind in the first place, and might generally be trying to maximize utility quotas other than fairness-related ones.

On that line, and somewhat predictably, not all sources of bias can always be traced back to a statistical or computational fault alone. There is a part of those unfairness sources that have their root in more subtle intrinsically human or systemic processes (Schwartz et al., 2022). Some solutions proposed for AI bias issues might sometimes miss the mark when it comes to addressing these and, therefore, also fail to adequately grasp the equally indirect and rippling impacts they can have within the larger social context in which they operate.

The more complex systems are, the more are the chances of accruing intertwined layers of bias, at multiple scales and by different entities, which leads to simpler attempts to analyze empirical data becoming more difficult to interpret, and to reliably pinpoint its causation. Hence, some academic efforts have turned to simulation to better study these complex models in a more accessible and adaptable manner (Du et al., 2021; Momennejad et al., 2019). By using a simplified abstraction of these micro- and macro-systems that influentiate us, one can entertain experiments that would be too large or take too long for empirical study, as well as study how they would evolve and feedback over time, in a more affordable way.

Thus, by using simulated environments, automated decision-making systems can be trained and probed in a more controlled and controllable manner, while avoiding some of the shortcomings empirical data based experiments do encounter. Indeed, a great benefit in this sense is that decisions and their outcomes can be considered for their future impacts in addition to their more immediate consequences, and are these long-term, potentially cumulative, effects that we can find when experimenting on a non-static environment that changes with time according to said decisions, which this project is mainly concerned with.

Specifically, in this work, we model the relation between an automated-decision making system and its environment as a *Multi-Armed Bandit* (MAB) problem. Then, we will want to examine and address both how the model behaves, as well as its consequences over a set of sensible features present in the environment.

Multi-Armed Bandit problems are characterized by a learner agent trying to maximize the total expected reward of its actions, dictated by an evolving policy, over an environment that presents multiple options to the agent, each with an initially unknown reward distribution. The solution to such problems can be summarized roughly as how many times the agent will take non-optimal actions before collecting enough information to settle for the best option or most-rewarding “arm”, and which arm will that be and whether it is indeed the best possible one. This model can be applied to automated decision-making instances in the hopes of learning how to optimize the best outcomes and take the best decisions but, predictably enough, the criteria they might find most desirable to maximize might depend on underlying biases and end up disregarding equally valid options because of unfair reasons.

There is a special version of MAB problems, described as *non-stationary*, in which the reward distributions of each arm are not fixed and can change over time, so that the perceived best option at a given point might not be the best option in the future. Furthermore, a relation between the decisions or policy being enacted by the agent and the way in which these distributions will change over time can be made so that the agent directly affects the environment around it.

This paradigm more closely represents the fluidity aforementioned that any decision-making process can and will precipitate in the context in which it operates. Unfortunately, and similarly as well, any bias unfairly exploited by a careless MAB solution, as descrived earlier, could lead to progressively even worse and more unfair outcomes over time by affecting it with its discriminatory actions in a self-promoting manner, or even generate new discriminative behaviors from the explotation of options marginally "better" than other increasingly belittled ones.

The present work, then, is mainly interested in the phenomenon of *bias* and its *cumulative effect* which, when learned from and when taking decisions in the wild, could lead to the perpetuation or incremental escalation of the differences it imposes over a given domain.

The aims of this project are:

- Modeling of an automated decision-making system as a *Multiple Armed Bandit problem* where instances grouped based on sensitive variables present arbitrarily different reward distributions that end up being unfairly optimized and reveal bias.

- Study of the *cumulative bias effect* that said model gives rise to when its decisions have an effect on the environment, which creates feedback loops that expand any existing discriminations over time in a manner consistent with real-life scenarios.

- Proposing solutions to MAB problems where *sensitive features* exist within the data which consider not only the exploration-exploitation dilemma but also take into consideration *distances between distributions and arm commitment*.

The goal is for all experiments to be able to showcase general trends applicable to a broader range of AI-decidable, bias-susceptible, domains. However, in order to base our efforts into a reasonable set of basic assumptions and collect relevant, interpretable insights from the exercise, the aforementioned decision-making model was presented with data inspired on trends extracted from actual US mortgage lending databases (FFIEC, 2020).

The separate and combined effect of two major types of bias (i.e. sampling and label bias), detected on the original data and adopted in turn by the algorithm, itself modeled to cause repercussions on posterior iterations of the experiment, was indeed found to lead to emergent confounding cumulative bias.

When designing the pack of solutions that would constrain the model into impeding this phenomenon from happening so intensely, or altogether, it is important that intersectional fairness metrics are recollected and appropriately used. Such approaches could include a method to make sure said biases never infiltrate the training pipeline to begin with or, perhaps more reasonably, in the case of this project, a plan to consistently review and enforce that these metrics are kept under *acceptable values* all throughout its learning and deployment stages.

Four original constraints are proposed and examined as to how successfully they are able to properly account for the established unavoidable impacts the AI's decisions will have moving forwards, and act accordingly to frustrate the rise of cumulative bias.

2. BACKGROUND NOTIONS

Useful knowledge will be presented and explained as a touchstone for our developments all throughout this work. These will consist in a more thorough characterization of bias as we will understand it, and an introduction to the MAB frameworks we use.

2.1 Bias

We will clarify what we refer to when considering AI bias an issue, and we will describe some classes of bias relevant to know for the purposes of this project.

a) Definitions of bias

Bias as a concept is crucial for the adequate performance of AI. Naturally, we usually want significant relations between variables to be made so that both supervised or unsupervised ML tasks can be carried out successfully over a given input dataset. This means disentangling how features are related or differ from each other, in order to use this information when classifying the data points or predicting probable outputs.

We can use *"bias"* to refer to how these given or learned relations between parameters point to a given sample leaning toward one class or another. This is expected to be based on some assumptions that can not always be true when dealing with data not present in the original training dataset, and this gives rise to the so-called bias-variance tradeoff one must keep in mind when designing most models (Lattimore and Szepesvari, 2020). In order to be able to account for unseen data and generalize correctly, we must *accept a degree of error* between some of the true values and estimated ones, or between the relations we would expect to be ascertained and the ones actually found.

A system with high bias and low variance will generally pay less attention to the training data and might end up missing some of the more subtle relevant relationships among the features, oversimplifying the model. This can be caused by an underfitting issue. Alternatively, a model with higher variance and low bias will make less assumptions and retain more complex information about the available data, but brought to an extreme this attention to detail can lead to worse performance on unseen data because of failing to detect useful patterns due to an overfitting problem. As stated, this tradeoff is ever present and ever relevant, but at the end a dilemma to be expected to have to face, and tolerate to an extent.

An even less problematic use of the term *"bias"* can be found, for example, when we need to express how certain trends do not pass through the origin in a regression context, so a bias term needs to be added to translate a given function to the relevant domain or range. Also, in Neural Networks, learned biases are absolutely essential to determine when a given node will be fired, shifting its activation function as necessary.

All these meanings of the word, however, with statistical, mathematical or scientific connotations, have little to do with the more harmful definition of bias we aim to showcase and address in this work. *Ethical bias* in AI refers to a wider range of issues related to trust, privacy, fairness and accountability concerns, that come from the design and use of underdeveloped systems that contain detectable lacks or prejudices in their sense-making models (Schwartz et al., 2022). Such biases can be introduced in any or every stage of their development, starting from the data creation step, to the problem formulation, the analysis of said data, or the validation and testing of their final solution implementation (Srinivasan and Chander, 2021).

b) Types of bias

Some of the most usual and relevant to this work types of bias are:

Definition 1. (Sampling Bias) One of the most frequent types of bias, it originates when creating a dataset with more instances of one type than others, rendering it under-representative of the domain it tries to capture. It can result in poor generalization and undermine the relevance of a given group within the data.

Definition 2. (Negative Set Bias) When building a dataset for a classifier, it is important to ensure we are defining any given phenomenon with positive instances about "what it is", but also with negative instances of "what it is not". Not doing so enough can cause poor performance in detecting those negative instances.

Definition 3. (Measurement Bias) It is introduced not when selecting samples for a dataset, but instead has to do with the data these samples are made of. It may have to do with errors during the *capture* process (because of intrinsic habits or human mistakes), with the *device* used to take the measurements, or happen because of *proxy* variables being used instead of the true features meant to be considered. For example, arrest rates can often be used as a substitute for crime rates, yet they do not convey the same information at all.

Definition 4. (Label Bias) Subjective preferences or criteria followed by annotators during the labeling process can introduce this type of bias to the training or testing stages, and it is easy to realize how this will affect the model learned. If the true labels are wrong or inconsistent to begin with, the model does not stand a chance in escaping this bias.

Definition 5. (Confounding Bias) This bias originates when the algorithm fails to grasp relevant relations that exist between features if common causes affect both inputs and target outputs. This can take several forms, from having key variables missing from the analysis altogether, originating *Omitted Variable* bias, or perhaps most frequently by creating a second type of *proxy* bias where indicators implicitly represent features other than themselves. For example, zip code might be also indicative of race, as people of a certain race might predominantly live in a certain neighborhood.

2.2 Multi-Armed Bandits

Decision-making with uncertainty is a common challenge we all face in many situations, and Multi-Armed Bandits do provide a simple formulation of this problem.

In a Multi-Armed Bandit problem, a *learner* agent is faced with an *environment* consisting of a set of possible actions, or "arms", that will return differently distributed rewards each time they are chosen. These distributions are initially unknown by the agent, who will have a determined number of rounds, or *pulls*, available to them in order to ascertain which arm is the most preferable one, and accumulate the highest total reward possible over time. We quickly realize how this premise will give rise to the well-known exploration-exploitation dilemma in Reinforcement Learning.

The policies developed by the learner over time, and the actions it will take in each round, depend on a history of past actions and perceived rewards.

Then, we'll evaluate these learners relative to a given policy by measuring their *regret*, or cumulative difference between the total expected reward of using the policy and the total expected reward collected by the learner over a given period of *T* rounds, also called *horizon*. Since we're usually interested in the performance of the learner in the long run, where chance will have a smaller effect in the comparison between two or more policies being put to the test, we want to look at the growth rate of the regret as *T* grows.

a) Stationary Stochastic Bandits

Stochastic Bandits are an elemental type of Bandit problems where for a limited amount of rounds T, each round $t \in [1, T]$ the learner will choose one of the *k* available actions $a_t \in A$ and receive a reward $X_t \in \mathbb{R}$ sampled from distribution P_{a_t} by the environment.

Very simple algorithmic approaches to navigate an environment such as these are described by the *Explore-Then-Commit* (ETC) or the *εpsilon-greedy* algorithms (Lattimore and Szepesvari, 2020). However, another well known algorithm with a more elaborate approach to the exploration-exploitation dilemma, characterized by the addition of a level of confidence on any of the present reward perceptions, is Algorithm 1 - *Upper Confidence Bound* (UCB), and is based on the principle of *"optimism in the face of uncertainty"* (Lattimore and Szepesvari, 2020).

The fundamental principle of UCB can be summarized as recommending for one to act as if the environment was as favorable as plausibly possible. Being *optimistic* when confronted with lack of data favors exploration, because we assume that untried options can be the best yet. Taking a *pessimistic* approach, on the other hand, would discourage exploration and one might suffer notable regret if this leads to missing out on unknown better options.

In contrast to *εpsilon-greedy*, where we perform exploration by selecting an arbitrary action chosen with a probability that remains constant, UCB algorithms change their exploration-exploitation balance as they gather more knowledge of their environment. UCB moves from being primarily focused on exploration, when actions that have been tried the least are preferred, to later further concentrating on exploitation, selecting the action with the highest estimated reward. This preference for unexplored actions is expressed through a positive *uncertainty term* that is added to the *experimental mean* reward for each arm, used each round when choosing the next action so that:

$$a_t = argmax_{a \in A}[\widehat{\mu}_t(a) + l\sqrt{\frac{log(t)}{N_t(a)}}], \quad (2.1)$$

where:

a_t - action selected at round *t,*

$\widehat{\mu}_t(a)$ - empirical mean reward of action *a* at round *t,*

l - confidence level chosen parameter,

$N_t(a)$ - times the action *a* has been selected by round *t.*

We can understand this operation as assigning an *upper confidence bound* value to each arm, based on the data so far, which will most likely be an overestimate of the unknown mean. This embodies the optimism principle previously stated. This term, however, which is multiplied by a given confidence level parameter *l,* does become less decisive as the amount of times an arm has been selected grows, progressively giving more credibility to its experienced mean reward alone.

There are many variations of the UCB algorithm, but this was the one chosen for this project's scope.

The intuitive reason why this upper confidence bound based criteria will lead to sublinear regret is as follows. Assuming that the upper confidence bound assigned to the optimal arm is indeed an overestimate, then another arm can only be selected if its upper confidence bound is larger than that of the optimal arm (which in turn is larger than the real mean of the optimal arm). And yet this will not happen too often because as additional data is progressively collected by playing any suboptimal arm, this will eventually cause their upper confidence bound to fall below that of the optimal arm.

Algorithm 1 - Upper Confidence Bound (UCB)

Input: T and l

Output: $\sum_{t=1}^{T} X_t$

while $t \leq T$ **do**

Choose arm $a_t = argmax_{a \in A} \left[\widehat{\mu}_t(a) + l\sqrt{\frac{\log(t)}{N_t(a)}} \right]$

Observe reward X_t and update $\widehat{\mu}_t(a_t)$ and $N_t(a_t)$

end while

This introduction to Multi-Armed Bandits already allows us to appreciate how further variations and considerations can be gradually taken into account to further expand the applicability of this type of problems to a wider range of cases, and find corresponding solutions, or useful insights, to each new version. In particular, one of these modifications is also key to know of for the purposes of this work, since it was absolutely influential to the approach finally taken.

b) Non-Stationary Stochastic Bandits

Non-Stationary Bandits are those problems that do not take for granted that the reward distributions behind each of their arms will remain constant, and must consider that they may change over time instead. Still, they are based on the assumption that they will change infrequently or drift slowly. For these Bandits, the aim of minimizing regret relative to the best actions in hindsight becomes somewhat unsuitable, because this "hindsight" notion becomes fuzzy and the whole point is to adapt to changes, in the optimal arm as well. For these, there exist two types of problem formulation: i) the moments when changes are applied are known; and ii) such information is unknown.

For the first case, applying a new UCB on each interval, or every time a change takes place, is a viable option enough to achieve a sublinear regret bound (Lattimore and Szepesvari, 2020, pg. 381 Eq. 31.4). This is the approach chosen in this work.

In cases when we don't know when changes take place, a *discount* factor can be added to the sampling formula we use to choose the next arm in order to give less weight to observations taken far in the past, which makes the algorithm most influenced by recent events. Alternatively, *sliding-window* UCB will directly discard observations eventually, after a given constant period τ. However, neither of these variations was actually used.

c) Other Bandit models

In addition to Stationary and Non-Stationary Stochastic Bandits, it also proved useful to learn about a few last special bandit models. Regardless, they were used only as inspiration, and won't be expanded upon as they weren't fully implemented.

Budgeted Bandits (Xia et al, 2015) are those that start with a fixed budget over a number of resource types, progressively depleting them according to some costs sampled from distributions that depend on each action taken. Their runs do not end after a fixed number of rounds, but instead do so when any of the budgets could become empty at the end of the current round.

Finally, there has also been an increasing interest in the development of Bandit learner agents for which the reception of feedback from the environment is not immediate (Lattimore and Szepesvari, 2020; Vernade et al., 2017; Vernade et al., 2018). They fall into the category of Partial Monitoring problems. This "learning with delays" model resonates well with how decisions taken in the present over a targeted number of neighborhoods or demographic groups might have an impact that might not reveal itself until considerably later. This, specifically, is also explored in some recent MAB research (Tang et al., 2020).

However, a common feature of a lot of the aforementioned Bandit problems is that the learner never needs to actually plan for the future. They might have to invest in exploration now to perform better later, or might have to continuously validate their knowledge over a changing environment, but in any case their available choices and rewards tomorrow will not be affected by their decisions today.

This is not true for the scenarios where the decisions of the model do influentiate its environment, in addition to the other way around. Similarly, most Bandit problems do not take into account that the environment might consist of agents with strategies and priorities of their own, which the learner would benefit from considering.

It is with the combination of all these theoretical frameworks, from which we can take inspiration and specific ingredients of each, that we might now feel more apt to approach the problem at hand.

3. PROBLEM

We will outline our reasoning behind the approach that was taken to identify and model the problem addressed in this project, we will make use of our background knowledge to formalize it as a MAB, and after describing the dataset that was used as a baseline for the creation our real-data based simulated environment, we will show indications on it of both bias and potential for cumulative bias effect.

3.1 Context on the Approach

The automated decision-making system chosen for this project is one in which a limited amount of desired resources or outcomes need to be fairly distributed among a larger than available number of candidate inputs. That is, the decision on whether a given input will get said desired outcome or not depends not only on the validity of its features, but also on the logistic considerations that come with having to manage a finite amount of resources over a possibly greater number of options, even if deserving.

Furthermore, and using the nomenclature suggested by Singh et al., 2022, *sensitive features* are assumed to be present within the data that describes these inputs which could, but should not, affect their prospects during this decision-making process. These sensitive labels, that define properties that directly or indirectly could identify samples as belonging to one or several protected groups, are discrete variables with a given amount of non-excluent *sensitive options* possible, and can be combined to result in the various intersectionally defined groups present in the data.

When learning about the configuration of the dataset to be decided upon, and when probing the environment for underlying reward distributions in order to pinpoint the optimal decision policy that will maximize them over time, it is reasonable to assume that not all applicants will present the same measure of favorable attributes that would make them a worthy candidate, and therefore not all will produce the same reward if given the chance. If they did, and were all identical in this sense, the decision-making process would seem to become trivial.

However, and based on the premise that these aforementioned sensitive features do not encode for relevant data in the decision-making process, we should hope that different sets of data grouped according to them wouldn’t show any significant difference in their respective reward expectations. Differences of perceived reward distribution should be related exclusively to differences in actual determinant factors in the data, and not on these sensitive parameters.

If this is not true for the initial data from which the system is supposed to learn, we cannot guarantee that the aspects it will learn to optimize do not do so on top of unfair underlying biases that it will ultimately help to maintain and expand all throughout its execution. Because of this, while it's valid to aim to optimize these determinant features in order to produce the best outcomes possible, that shouldn’t happen at the detriment of equally eligible applications marginalized by the *system*, inside and outside of the automated decision-making model context.

A way or set of methods to make sure that the system is willing to accept a certain degree of suboptimality to account for these potential objectionable initial conditions is needed. That is, while sampling and label type biases, for instance, can and likely will be detectable to some extent in the data to be processed, appropriate mechanisms and metrics may be used to mitigate their effect in the long run. Especially considering that, alternatively, if not addressed at all, we can assume that they will remain or even progressively grow instead.

3.2 Formalization as a Multi-Armed Bandit Problem

Let us now use all this knowledge and background in order to formalize the problem we are trying to define and deal with. Our agent will be presented with data samples C over which they'll have to take a binary decision of granting a given resource. However, the amount of resources our agent can give out are limited and defined as $C' \subset C$. Each instance $c \in C$ consists of a series of features $\Phi(c)$, some of which we'll consider to be *sensitive*. That is, they are variables that aren't used in the problem optimization when making said instance more worthy of one outcome or another, but instead help to describe it in terms of protected minorities in which it could be considered to belong.

$$\Phi(c) = \Phi_{sensitive}(c) \cup \Phi_{\neg sensitive}(c) \tag{3.1}$$

$$\Phi_{\neg sensitive}(C'') = (\Phi_{\neg sensitive}(c) : c \in C'' \subseteq C) \tag{3.2}$$

If we consider we have *s* sensitive discrete parameters within our data, each parameter *i* with a number o_i of possible *sensitive options*, we can express the amount of sub-groups the data can be divided on, by joining instances with the same combination of sensitive attributes, like so:

$$k = \prod_{i=1}^{s} o_i \tag{3.3}$$

By having our data divided into these *k* subgroups or classes C_a, each representing a different demographic susceptible of being present in our data, we have that the total size of our data, expressed as $|C|$, is the sum of the sizes of each of these subgroups:

$$|C| = \sum_{a=1}^{k} |C_a| \tag{3.4}$$

In the context of a Multi-Armed Bandit problem, we will represent each of these k groups as a different arm *a.* For each action $a \in A$, we have a reward distribution $P_a \in P$ from which the environment will sample the agent's reward every time said arm is selected.

These reward distributions P_a, represented as Γ in Formula 3.5, could be inferred from supervised data by extracting the probability distributions of samples from each group being attributed any kind of qualitative valuation or binary label, or be determined by any pertinent function f in order to be based on the concrete domain to be analyzed.

$$f : \Phi_{\neg sensitive}(C'') \rightarrow \Gamma \tag{3.5}$$

$$P = (f(\Phi_{\neg sensitive}(C_a)) : a \in A) \tag{3.6}$$

In our case, each arm will also have a finite amount of pulls available $n_a \in N$ depending on how many instances of that group are present in C. Then:

$$N = (|C_a| : a \in A) \tag{3.7}$$

At every round $t \in \{1,\ldots,|C'|\}$ the learner will choose one $a_t \in A$ and receive a reward $X_t \in \mathbb{R}$ from distribution $P_{a_t} \in P$.

The motivation behind the choice of representing each demographic as an arm with potentially differently distributed rewards responds to the fact that, instead of being presented with data points in a sequential way, since we need to manage a limited amount of resources, we should be able to better allocate them by working in batch-like fashion and being presented with all options at once. This way, the learner will learn to prioritize the best arms and exploit them as much as they are available to them, and then keep progressing through the next best options until being left out of more resources to grant.

A solution to the problem would be the sequence of actions that brought about the highest expected cumulative reward possible.

However, this reward optimization process that UCB will enact, if done carelessly, can result in unwanted discrimination toward some of our protected groups since underlying assumptions or biases present in our inital data (e.g sampling or label bias) could imbue such negative effects or repercussions to *P*, *N*, and therefore to our optimized solution. We'll elaborate more on this later in Section 3.4.

3.3 Introduction of the HMDA Database

Now we will be presenting the real-data US mortgage lending dataset that is adopted as a baseline for how biases can indeed be found in our input data. A setting similar to the one used in the DualFair de-biasing pipeline showcasing paper by Singh et al., 2022, was procured from the mortgage lending applications database by *The Home Mortgage Disclosure Act* (HMDA).

Indeed, we chose mortgage lending as a known use case of the problem we described, in which finite resources must be distributed over a demographically diverse population, and which has proven to be susceptible to bias, to dire effects (Weber et al., 2020). In our case, we imagine that a given financial institution may want to decide on the overall frequency of loan applications it will approve from different social groups. We consider this a sensible premise as it is also contemplated in Tang et al., 2020.

We made use of a starting array of 1,557,705 samples from the US states of Alabama, Arkansas, Georgia, Mississippi, Louisiana and Tennessee produced during 2020. The data consists of 99 features and a single binary target label which encodes whether the loan originated or was declined. Within the data, 3 separate sensitive features are observed and correspond to the derived *race, ethnicity* and *sex* of the applicants.

In our simplification of the problem, the options considered for each protected variable are race={**W**hite, **B**lack or African American}, ethnicity={**N**ot **H**ispanic or Latino, **H**ispanic or Latino} and sex={**M**ale, **F**emale}. Therefore, and differently than in the DualFair case, each sensitive feature will have cardinality 2, losing the “Joint” option for each, that originally represented the instances in which the applicant and co-applicant of the applications were of different options.

After discarding all samples that represented options other than these six, we ended up with 717,997 data samples instead.

As described in Formula 3.3, the cross product of each of the 3 sensitive feature cardinalities yields to the number k of all possible sensitive groups. By grouping the dataset into subsets sharing the same combinations of those three sensitive attributes, we end up with 8 different demographic groups present in various degrees within our final dataset. Keeping "Joint" as an option would have brought that amount to 27 groups instead, further encumbering our analysis.

We can visualize the magnitude in which each group is unequally represented in the data in the following plot, as well as how many of its applications were approved or not:

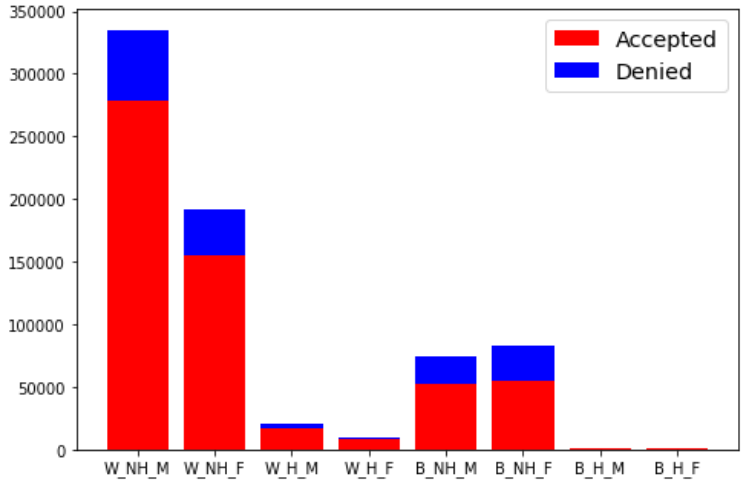

Figure 3.1: Inspection of instance distribution in HMDA database

By continuing to use this 8-groups based analysis, we can similarly closer examine the approval rate of applicants belonging to each class.

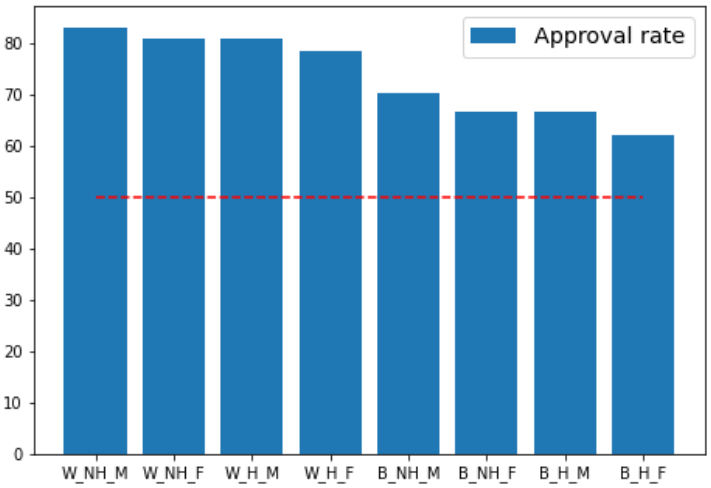

Figure 3.2: Inspection of approval rate spread in HMDA database

We can already appreciate some suspicious differences in Figure 3.2 among the rates of the demographics we would expect to be generally more favored (e.g. **W**hite, **N**on-**H**ispanic or Latino, **M**ales on the far left), and those somewhat less privileged (e.g. **B**lack, **H**ispanic or Latino, **F**emales on the far right). However, another aspect we can learn from our data by looking at both these plots is the fact that, for all groups, the approval rate seems to be considerably high.

Getting back to the 8-groups analysis, we can put these approval rates in context to see if these differences translate well to the fact that groups are also differently populated, as made evident in Figure 3.1.

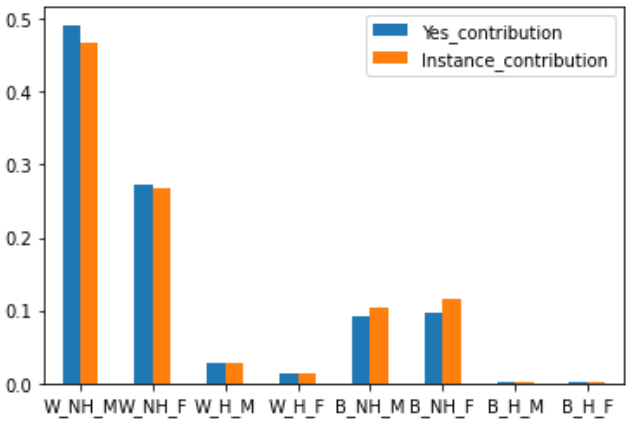

Figure 3.3: Contribution of each demographic to data and to total approvals

Indeed, groups on the left make up for a higher approval proportion over the total number of instances approved than their own contribution and presence in the data. On the other hand, groups further on the right seem to more generally fall under that same mark.

With this information we can already identify some hints of bias being manifest in the data. We can talk about a serious instance of sampling bias in the distribution of each class over the total data, with some demographics appearing more than a 100 times more than others. These percentages should be contrasted with real demographic distribution statistics from the states where this data was collected from, to see if this is consistent with the real-world context or not, but in any case this imbalance might end up conveying to any algorithm that some classes are less important to consider or treat fairly than others, as they are less present overall.

Furthermore, the fact that some groups are overrepresented or underrepresented in the total amount of approvals could also be indicative of label or confounding bias, on top of the sampling issue. This will be elaborated on later in Section 4.1. Finally, we can talk of a possible case of negative set bias caused by having a much inferior amount of declined applications in the dataset compared to positive, originated ones.

In our environment, we will use the relative approval rates of each group as a base for our initial reward distributions' vector P. We are modeling each $P_a \in P$ as Bernoulli distributions, characterized by a term $p_a \in$ [0.83, 0.807, 0.808, 0.782, 0.702, 0.664, 0.666, 0.619], respectively, as extracted from Figure 3.2. Each arm a will sample their rewards from their P_a and return $100 * 1$ with probability p_a or 0 with probability 1 - p_a.

$$\mathrm{P} = (Bernoulli(p_a) : a \in A) \tag{3.8}$$

Also, we will use our insights from the HMDA data to weigh how input environment vector N, containing the limited amount of times each arm will be able to be pulled, will be created at every execution of the algorithm. The proportion information we got from our analysis in Figure 3.3 will describe these initial population weights $w_a \in W$, which add up to 1, and are W = [0.467, 0.267, 0.028, 0.014, 0.103, 0.116, 0.002, 0.002]. They will be used to inform a Multinomial distribution that will form N, further characterizing our 8 arms before each run of the algorithm.

$$\mathrm{N} = (n_a \sim Binomial(|C|, w_a) : a \in A) \tag{3.9}$$

3.4 Problem Showcase in a Stationary Setting

Using extracted parameters P and N to characterize the 8 arms representing each of our focused demographics, we let the UCB algorithm execute over a total of $|C| = 2500$ simulated data points each time, allowing $|C'| = 0.8 * |C|$ (i.e. 0.8 was chosen as it is an approximation to the actual overall percentage of approved applications), with a chosen confidence level of $l = 25$. The experiment was run 100 times, and the mean μ and standard deviation σ of the metrics evaluated at the end of each run can be displayed in the following way, mirroring Figure 3.2 and Figure 3.3:

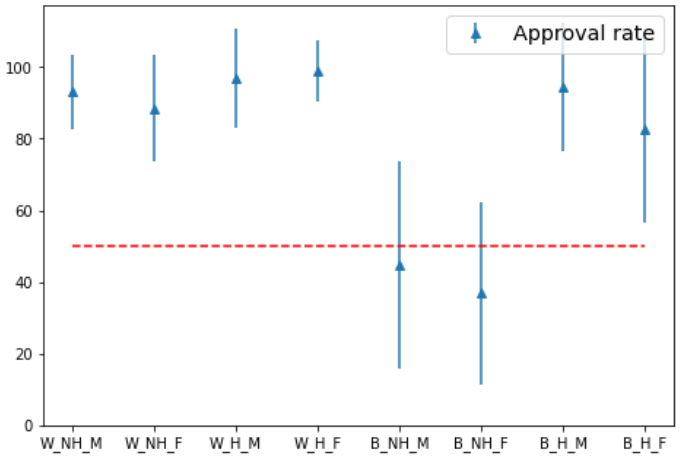

Figure 3.4: Approval rate granted by UCB per each demographic after 100 stationary executions on HMDA-based data

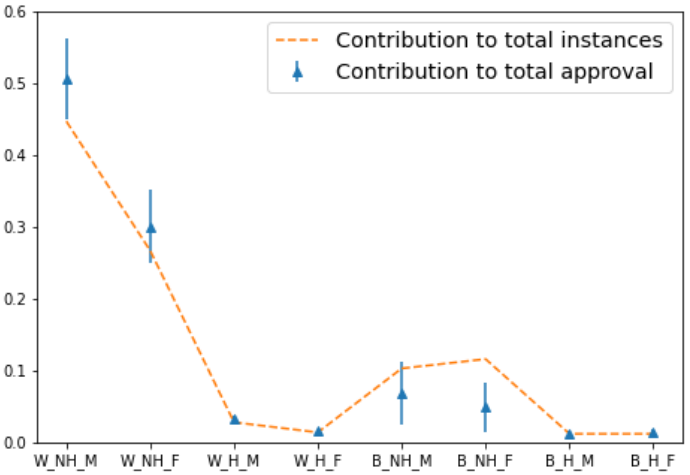

Figure 3.5: Contribution of each demographic to data and to total approvals after 100 stationary executions on HMDA-based data

We can compare these with the figures obtained when running the same experiment on a different environment where all eight arms share the same reward distribution and population proportion. That is:

$$P = (Bernoulli(0.5) : a \in A) \quad (3.10)$$

$$N = (n_a \sim Binomial(2500, 1/8) : a \in A) \quad (3.11)$$

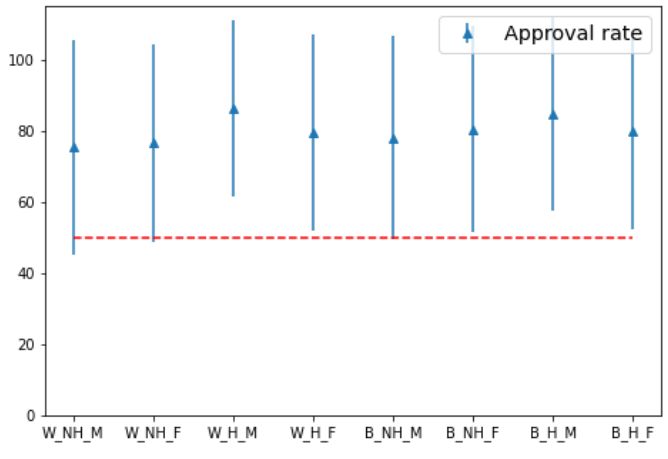

Figure 3.6: Approval rate granted by UCB per each demographic after 100 stationary executions on unbiased data

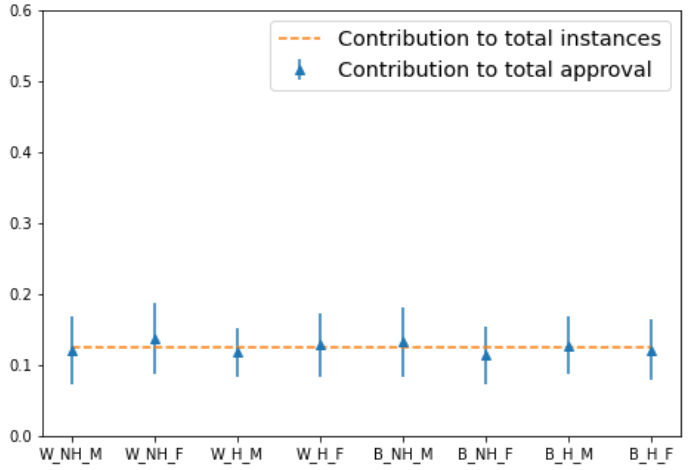

Figure 3.7: Contribution of each demographic to data and to total approvals after 100 stationary executions on unbiased data

The differences detected and extracted from the HMDA dataset do have an effect in the behavior of the algorithm, and we see that this effect's impacts are targeted toward some demographics more than others. That is, in Figure 3.4, we see that when choosing how to distribute 80% of the total pulls among all arms two in particular are being consistently left out of C'. Also, similarly to what we saw in Figure 3.3, in Figure 3.5 we can see how more privileged groups have a greater influence on the total number of approvals than they should, while other groups experience the opposite. This doesn't happen to the same extent in the unbiased environment.

This wouldn't necessarily be a problem if we didn't consider these differently regarded and selected groups as what we established that we wanted them to actually represent. Indeed, in our setting, we are using these arms as an abstraction of the demographic groups present in our input data. Their reward distributions come from the analysis of historical resolutions of each group's applications, which we should expect to not have to do with the sensitive features we use to build these groups, but be based on a function of their determinant features instead. Therefore, we have to assume that differences in these reward distributions, or differences in the selections' distribution after each run, do showcase and are caused by biases we will want to avoid. These optimizations then, although the algorithm is performing correctly, are unfair.

Moreover, until now we've been considering that the environment the learner had to probe was static. That is, that the rewards assigned to each arm or the composition of the input and proportion of each group within it didn't significantly change over time. Due to how the rewards are sampled from Bernoulli distributions in this modelization, and the population to be decided upon is chosen randomly using some initial weights, there was room for differences from one execution to another. Regardless, these differences aren't reflective of any progress by themselves.

3.5 Problem Showcase in a Non-Stationary Setting

Let us now observe how our Multi-Armed Bandit problem would change under alternative circumstances. We will be using the same UCB algorithm, but on this occasion we will establish that the environment is non-stationary instead. After each batch of $|C'| = 2000$ pulls to be selected over a pool of $|C| = 2500$, we'll enforce some changes on the environment to be sampled in the next iteration.

These shifts will take place in two key attributes of our model's context. Both the reward distribution as well as population weights of each group will evolve depending on the allocation of pulls in the previous batch. The assumption is that groups that prove to be more profitable will become increasingly preferable over other "less reliable" ones, and that applicants belonging to said groups will accordingly feel more confident and apply more often, occupying even larger portions of the data in relation to the other more discouraged demographics. These assumptions also stem partially from previous literature (Bartlett et al., 2022; Cowgill and Tucker, 2019).

The · symbol represents the dot product between vectors in the following formulas:

$$presence\_data = (\left|C_a\right|/|C| : a \in A) \quad (3.12)$$

$$presence\_approval = (\left|C_a \cup C'\right|/|C'| : a \in A) \quad (3.13)$$

$$increments = (1 + \frac{presence\_approval_a - presence\_data_a}{presence\_data_a} : a \in A) \quad (3.14)$$

$$reward\_d_{t+1} = (1 - r_{coef})reward\_d_t + r_{coef}(increments \cdot reward\_d_t) \quad (3.15)$$

$$population\_w_{t+1} = (1 - p_{coef})population\_w_t + p_{coef}(increments \cdot population\_w_t) \quad (3.16)$$

The combined effect of these two updates in-between executions can have drastic effects if starting from underlying differences that end up escalating over time. Next up is the recorded evolution of the two previous environments when being applied successive UCBs for batches of 2500 instances during 100 epochs. We used r_{coef}=0.01 and p_{coef}=0.1 to model the fluidity of both environments, which control how fast changes to the current environment's characteristics are applied, in order to perceive their consequence in a computationally reasonable number of epochs.

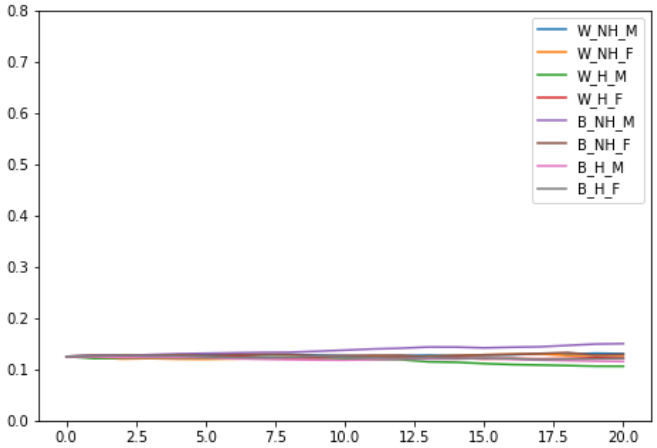

Figure 3.8: Evolution of population weights every 5 epochs in an initially unbiased environment

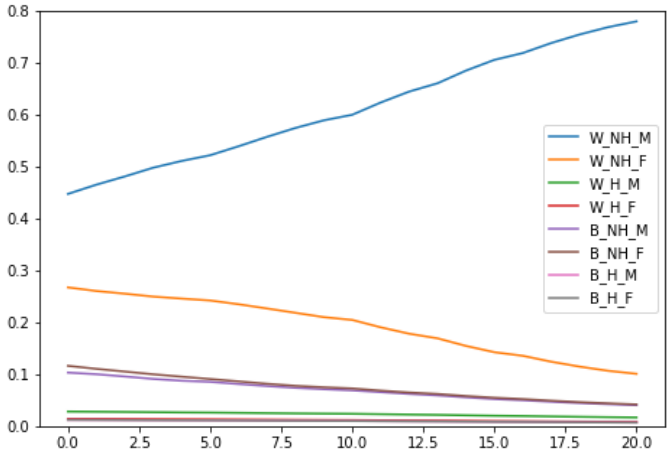

Figure 3.9: Evolution of population weights every 5 epochs in an HMDA-based environment

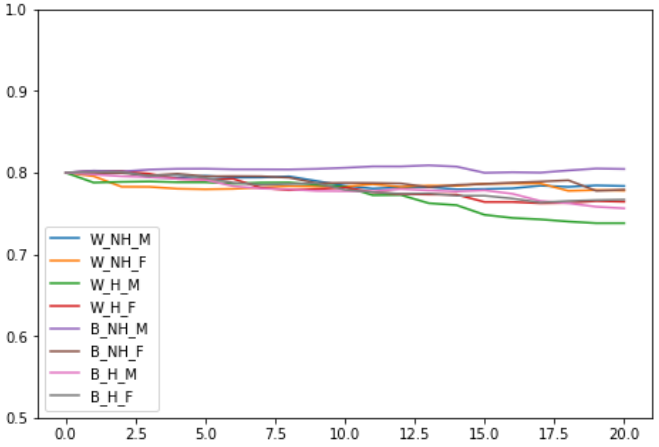

Figure 3.10: Evolution of reward estimates every 5 epochs in an initially unbiased environment

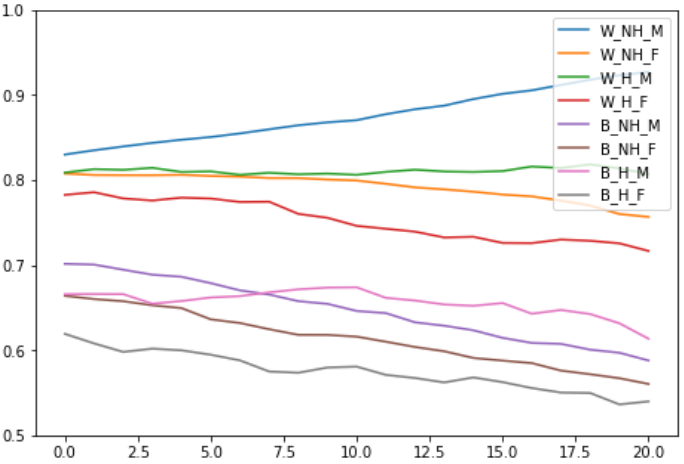

Figure 3.11: Evolution of reward estimates every 5 epochs in an HMDA-based environment

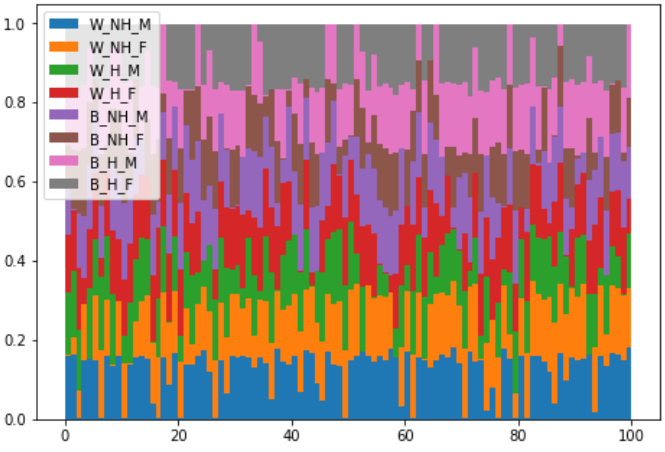

Figure 3.12: Evolution of contribution to total approvals for every epoch in an initially unbiased environment

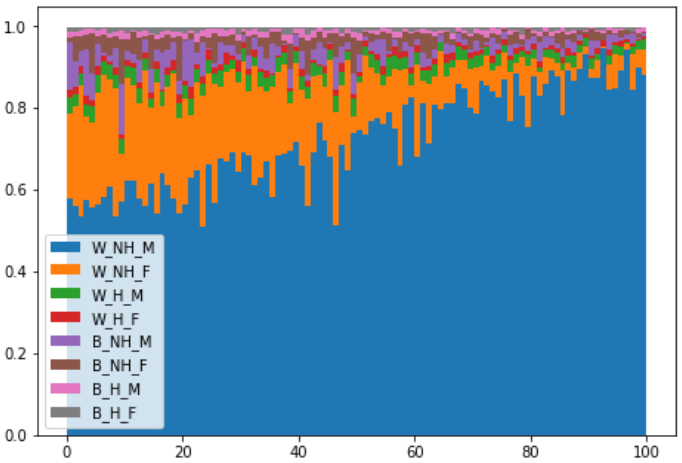

Figure 3.13: Evolution of contribution to total approvals for every epoch in an HMDA-based environment

In the unbiased environment we can see how the proportion in which each group is present in the data shifts slightly by the end of the 100 iterations, in Figure 3.8. Their reward estimates in Figure 3.10 perceive somewhat more pronounced changes, but also are maintained close to their original value for most of the experiment. Both evolutions are key to how each group's approvals contribute to the total number of approvals in a comparable fashion all throughout the 100 epochs, as seen in Figure 3.12.

Due to how UCB only devotes a finite amount of time to exploration, and how it is constrained to only grant 80% of all available pulls, each iteration some groups' reward distribution remain underestimated and become underselected in that particular iteration. However, since this happens depending mostly on the randomness involved in the process of sampling almost equally rewarded options, the model compensates for itself in the long run and the effects on the environment aren't immediate nor dramatic.

Still, this suggests how unadequate automated decision-making algorithms being left to their own devices and unchecked could still gradually cause bias and discrimination even on initially impartial environments.

Getting back to our HMDA-based case, however, we see how changes take shape much faster and much less arbitrarily. The group composed by **W**hite, **N**on-**H**ispanic or Latino, **M**ale individuals, which enjoys the highest initial value for both indicators, is given preference over the rest from the start which in turn leads it to accumulate favor in the model and environment and progressively further marginalize the rest of options.

By the end of the 100 epochs, the proportion and rewards of each group within the data have been affected so much by the unbalanced selection processes having been carried out that all other groups have become relegated to comparatively minor actors in the environment and selection process, as can be appreciated in Figure 3.9, Figure 3.11 and Figure 3.13.

We can now corroborate the full extent of the cumulative effect of those harmful initial conditions when being unfairly optimized and exploited over a extended period of time in a non-stationary environment like the one we modeled, which will expand them even further.

This is certainly a simplification of any comparable evolutive societal or economic process and needs to be considered as a dramatization of what would happen in a real-life scenario or a more sophisticated model. Nevertheless, by starting from the collection of conceivably reasonable assumptions stated throughout these developments, and working our way out from those, we have arrived at what we would undoubtedly call a fatal situation that we would wish to have avoided, and with no clear way on how to remedy it. Perhaps to a different degree or extent, it is still plausible to think that similar circumstances could precipitate a similar end result in our world, or already are.

In this sense, then, it is relevant that we approach this problem earnestly and try to find possible solutions to help us mitigate this cumulative bias effect we have now further identified.

4. METHODOLOGY

We will explain how we will tackle the problem we illustrated, define four bias metrics that we will use as constraints to UCB, and see how this will affect the implementation of the algorithm.

4.1 Context on the Approach

As established, we would still like to find a policy that will bring about the highest total reward possible by optimizing our data over our *non-sensitive* features. However, and like we have seen, doing this while not minding for our *sensitive* attributes as well can lead to this optimization being done to the detriment of the minorities within our data, due to being based on potentially biased underlying initial conditions. These biases will be incremented over time in a non-stationary environment, leaving us progressively even worse than when we started off.

We need to add certain constraints to our optimization function that will make it care for those sensitive features as well, and the implementation of which doesn't make us discard the better aspects of our chosen algorithm. Indeed, when it comes to solving Multi-Armed Bandit problems, considerable effort has been put into developing solid algorithms able to navigate the "exploration-exploitation" dilemma in initially unknown environments to adequately assess the most reliable option or decision from those available. In this case, we want to add a supplementary dilemma to be considered transversally during the execution of our updated version of UCB, in the suggested axis of "exploitation-equity".

For that reason, we'll introduce a different metric we'll also want to keep in mind during the optimization process, i.e. the bias being introduced by following any given policy. Since the sensitive features over which we have assembled these groups should not have any effect on the decisions taken over their instances, we actually would expect that their reward distributions aren't significantly different from each other. If they are, this would be indicative of underlying biases.

There are at least two types of bias that might have a hand behind this circumstance. It might be that there exists label bias if the annotators did indeed show some degree of prejudice when settling these applications originally, affecting the target labels the model would need to deal with. Alternatively, it's also possible that a combination of measurement and confounding proxy biases could have led to features from certain groups being globally better regarded and perceived as positive than others. That is, that the fact of belonging to these groups directly implies that their applications will be more likely to present a whole other range of features as well, "good" or "bad".

4.2 Formalization of Bias Metrics

Indeed, we can think of other multiple indicators of suspicious relationships among these groups that would prompt for mistrust in the criteria being followed, after a batch of decisions is presented to the model. Here is how we will formulate four of these.

1) The percentage of a groups' presence within the data and its difference to the percentage of approved instances from that group over the total amount of approved instances. Ideally, they should be the same within each group, that is, each group should contribute to the approval corpus as much as they contribute to the overall data.

$$presence\_data = (\left|C_a\right| / |C| : a \in A) \quad (4.1)$$

$$presence\_approval = (\left|C_a \cup C'\right| / |C'| : a \in A) \quad (4.2)$$

$$bias\_presence = \sum_{a=1}^{k} (\left| presence\_approval_a - presence\_data_a \right|) \quad (4.3)$$

The $\bar{v}$ symbol represents the average of the vector v it is applied to, in the following formulas:

2) The approval rate of each demographic group, relative to itself, should be the same and shared by all of them (preferably close to 0.8, in this particular case).

$$approval\_rates = (\left|C_a \cup C'\right| / \left|C_a\right| : a \in A) \quad (4.4)$$

$$bias\_rates = \sum_{a=1}^{k} (\left| \overline{approval\_rates} - approval\_rates_a \right|) \quad (4.5)$$

3) The posterior probabilities of an approved application belonging to any and all of the demographic groups. Ideally, we should perceive the same odds for each.

$$posteriors\_groups = (P(c \in C_a | c \in C') : a \in A) \quad (4.6)$$

$$bias\_posteriors = \sum_{a=1}^{k} (\left| \overline{posteriors\_groups} - posteriors\_groups_a \right|) \quad (4.7)$$

4) Finally, and as mentioned, the reward distribution of each arm P_a, able to be modeled by a mean μ_a and deviation σ_a, should also be indistinguishable from one another. If they aren't, even if this would not be the model's fault in a stationary environment as it wouldn't have any effect on them, this would still be indicative of some bias in our setting. The Kullback–Leibler (KL) divergence metric can be used to quantify the distance between two probability distributions (Ji et al, 2022). In this case, we would have to apply it among each pair of distributions "P" and "Q" and use the sum, average or maximum of those differences as our final *bias_rewards* indicator.

$$KL(P||Q) = \int p(x) log(\frac{p(x)}{q(x)}) dx, \quad (4.8)$$

for the case of continuous r.v.

As we are using Bernoulli distributions for our simplified modelization of this problem, we can think of a correspondingly simplified alternative to this calculation. Since we don't have access to the actual reward distributions of the environment from the scope of our algorithm, we can use the perceived empirical means of UCB and encourage it to give further chances to those groups that are currently regarded as worse than others, in an extension of its "optimism in the face of uncertainty" principle.

$$bias\_rewards = \sum_{a=1}^{k} (\left| \overline{\widehat{\mu}_t} - \widehat{\mu}_t(a) \right|) \quad (4.9)$$

Exhaustive and readily applicable bias and fairness metrics themselves can prove difficult to define, as well as to satisfactorily navigate the trade-offs that might arise even between different fairness definitions, and evidently between fairness metrics and other objectives (Silberg and Manyika, 2019). However, these constitute our attempt at it, for the purposes of this project.

4.3 Implementation

Using any of these definitions of bias, or a linear combination of all, as *bias_metric* $\in \{bias\_presence, bias\_rate, bias\_posteriors, bias\_rewards\}$, we could advise a different policy in contraposition to the one enforced by UCB such that it would prioritize the minimization of *bias_metric* over any other factor. However, this new policy wouldn't actually frame any notion of rewards as being something necessarily positive or desirable, by the model, or being susceptible of being more or less deserving of, by the data.

By defining and using some coefficient $\theta \in [0, 1]$ we are able to not lose all this relevant information about our environment and problem setting, while also updating the UCB algorithm into Algorithm 2 - *Bias Constrained UCB* (BC-UCB), in order to account for these objectionable differences and biases as well, up to an extent of our choosing. That is, we will enforce a *suboptimality degree* that the model must be prepared to accept and abide by.

We will include the influence of our new metrics in the sampling step by altering the sample values being produced, punishing those decisions that would further expand the differences, or bias, existing among the different arms.

We would no longer choose the next pull using Formula 2.1; instead, the new selection criteria will be:

$$a_t = argmax_a\,(1 - \theta) * [\widehat{\mu}_t(a) + l\sqrt{\frac{log(t)}{N_t(a)}}] - \theta * bias\_term_t(a) \qquad (4.10)$$

Where the *bias_term*$_t$ *vector* could be any current measurement we saw fit to counteract the drift of a reward-focused policy only. The way in which we implement it, using the *bias_metric* functions introduced earlier, is as described next.

During the sampling process, we will use one of our proposed *bias_metric* when influencing the UCB pure decision by a degree θ. We calculate how each action *a* would hypothetically affect the chosen metric on the next step $t + 1$, and then min-max normalize the vector of these hypothetical future bias values to the range [0,1], where 0 corresponds to the arm whose pulling would produce the smallest new *bias_metric* output value, and 1 corresponds to the highest.

Before being introduced in the final Formula 4.10 to be optimized, though, we multiply these normalized values by 100, a magnitude comparable to the range of the original UCB sampling terms.

$$MinMax(v) = \frac{v - min(v)}{max(v) - min(v)} \qquad (4.11)$$

$$bias\_term_t = 100 * MinMax(bias\_metric(t + 1)) \qquad (4.12)$$

Algorithm 2 - Bias Constrained UCB (BC-UCB)

Input: T , l , θ, $bias\_metric$

Output: $\sum_{t=1}^{T} X_t, bias\_metric(T)$

while $t \leq T$ **do**

Compute $bias\_term_t = 100 * MinMax(bias\_metric(t+1))$

Compute $A' = (a \in A \text{ s.t. } N_t(a) < n_a)$

Choose arm $a_t = argmax_{a \in A'}((1-\theta) * \left[\widehat{\mu_t}(a) + l\sqrt{\frac{\log(t)}{N_t(a)}}\right] - \theta * bias\_term_t(a))$

Observe reward X_t and update $\widehat{\mu_t}(a_t)$ and $N_t(a_t)$

end while

Like so, we can describe the way in which we will constrain our algorithm toward mitigating its negative effects over the environment just by defining the *bias_metric* function we'll use and a single suboptimality coefficient θ. These are the two hyperparameters we'll be evaluating our solution over. On the other hand, we'll consider the outputs of this algorithm to be the *final reward*, or total cumulative reward, obtained by the end of its execution, as well as the last result of the *bias_metric* used, also as computed at the end of its execution.

In our experiments, in addition to all four bias metrics, we will also record the *final reward* and a *final regret* metric we'll define as the difference between the mean reward produced by the optimal policy and the expected reward of all actioned pulls. We can simplify it to reduce its magnitude while maintaining the same correlation of values.

$$R = T\mu^* - E(\sum_{t=1}^{T} X_t) \tag{4.13}$$

$$R = \mu^* - E(X_t) \tag{4.14}$$

$$R = [100 * max(p_a)] - (\sum_{t=1}^{T} X_t)/T \tag{4.15}$$

In order to find the optimal θ^* in each case as to perceive the best combination of reward and bias mitigation results, the new measurement φ will be calculated and optimized:

$$Frobenius(v) = v / \sqrt{(\sum_{i=1} v_i^2)} \tag{4.16}$$

$$\varphi(frew, fbias) = Frobenius(frew) - \beta * Frobenius(fbias) \tag{4.17}$$

$$\theta^*_{bias\_metric} = \underset{(\theta)}{argmax} (\varphi(BC\text{-}UCB(\theta, bias\_metric)) \tag{4.18}$$

We use a *Frobenius* normalization function both to convert our *final reward* and final bias magnitudes to a comparable range, as well as to get a greater appreciation on how spread or significantly variate within themselves they are.

β can be used to introduce a notion of meta-suboptimality to enforce at this stage, or how important we consider the bias term in relation to final reward when determining the best value of θ. However, we used β=1 for our aims, considering that we value them equally at this layer of the variable-tunning process.

Indeed, φ could be computed differently, to the same aim of balancing both opposite objectives, yet this was the definition that was used when finding each "best" θ on the stationary experiments, in order to apply them later in the non-stationary scenario.

5. EXPERIMENTS

We will use our updated version of the UCB algorithm, BC-UCB, to process batches of simulated input data and evaluate its performance within the context of a single execution, during which we consider the context to be stationary. We will do so with each of our four introduced bias metrics as the constraint being enforced. Then, after extracting from these experiments the four θ^* that will reduce the relevant bias metric the most in each case without diminishing our rewards more than required, according to Formula 4.18, we will examine the behavior and consequences of the algorithm and these θ^* values when applied to an evolving, non-stationary, environment.

The detailed figures resulting from these experiments can be found in the Appendices section of this work, and they will be interpreted on these next few pages. Similarly, the code and functions used to carry them out can be found in the public repository https://github.com/quimHM/QHM_TFG_repository .

5.1 Considerations on Observations in a Stationary Environment

First, we'll analyze the isolated application of all four bias measurements as the single metric we use to calculate $bias\_term$, while still recording their effect on the rest as well, when applied in a non-static environment. In Appendix A, we showcase their mean μ and standard deviation σ values over 50 iterations with 2500 data points each.

By examining how each metric behaves depending on the degree of suboptimality applied in each case, we realize that they are implicitly related to each other quite considerably. Even though we are only using one metric in the sampling formula of the algorithm at a time, we notice how all of them are accordingly reduced at the same time, to various degrees.

We also can observe how they converge to their minimal values at different points, some quite before reaching $\theta=1$. It is relevant to keep in mind that these bias constraints do affect the normal behavior of UCB in more ways than one. In addition to the actual second term added to the sampling formula presented earlier in Formula 4.10, they also can affect, in turn, the way in which UCB's estimates are updated and therefore both terms are more intertwined than initially apparent. This impact can indeed coalesce into the UCB term also evolving to contribute in the bias mitigation on its own, causing the algorithm to converge before $\theta=1$, as seen.

Final reward and final regret, inversely related, also share common trends with the rest of metrics, as a connection among all of them clearly exists and was made manifest, whereas before we could only presume it.

Indeed, *bias metrics* and *final reward* decrease together, which we can interpret as that abiding by bias constraints has a cost, and it's generally more rewarding to exploit differences than to make efforts toward bridging them, in this case.

It is worth noting that an unexpected consequence of our method of computing φ was the fact that all perceived final rewards resulted in very close normalized values. This revealed that the relative difference in the rewards to be gained seemed much less significant than the improvement to be had in the biases measured, for different values of θ. This wouldn't necessarily be the case if constraining the model to generate less bias did have a bigger impact on the rewards produced. Surprisingly enough, the cost of doing so was less than anticipated. When subtracting the normalized values of rewards and bias metrics, therefore, the latter term became much more determinant in the shape of φ:

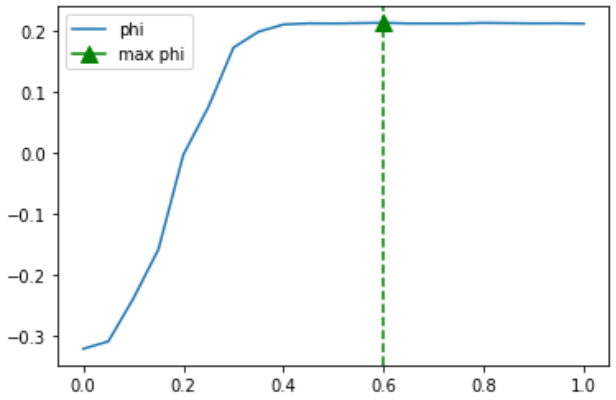

Figure 5.1: φ when *bias_metric* = *bias_presence*

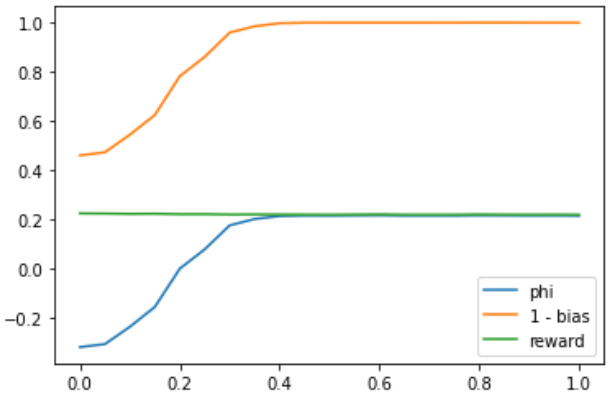

Figure 5.2: Normalized values of A.1 and A.3

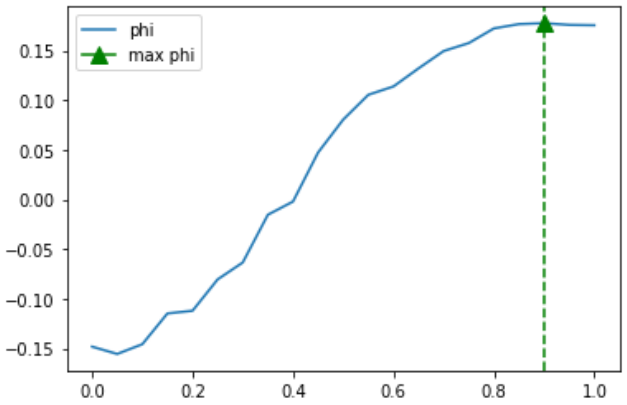

Figure 5.3: φ when *bias_metric* = *bias_rate*

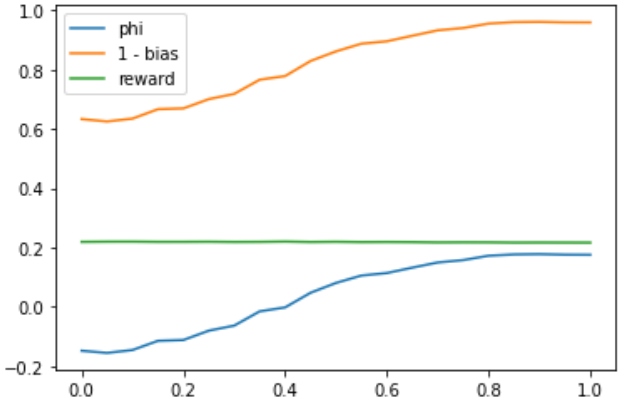

Figure 5.4: Normalized values of A.7 and A.10

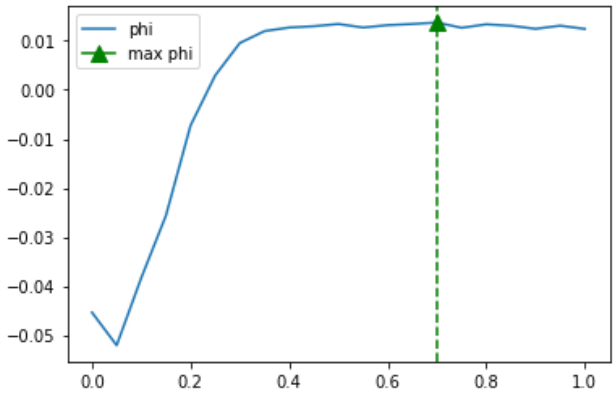

Figure 5.5: φ when *bias_metric* = *bias_posteriors*

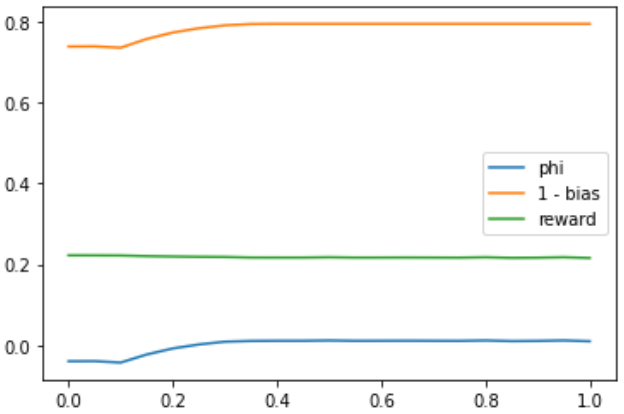

Figure 5.6: Normalized values of A.13 and A.17

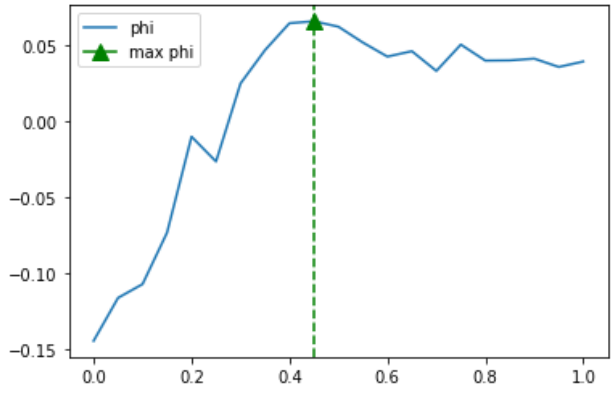

Figure 5.7: φ when *bias_metric* = *bias_rewards*

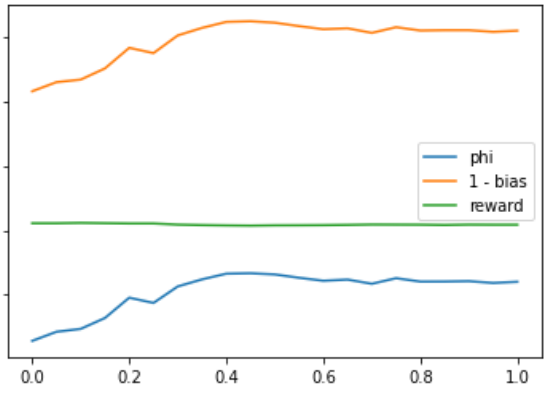

Figure 5.8: Normalized values of A.19 and A.24

5.2 Considerations on Observations in a Non-Stationary Environment

After putting all of our four constraints to the test, in Appendix B, while using in their application in a non-stationary setting what we detected to be the best suboptimality coefficient for each, i.e the θ that would reduce the relevant bias the most without lowering our rewards more than needed, we can draw some valuable observations:

Presence bias constraint: The bias term dominates the execution completely throughout, and by enforcing that each group is represented in the total amount of approvals in the same proportion that they exist in the data, it achieves exactly that. Since the differences in this metric, positive or negative, are actually what the environment is modeled to react to in order to evolve, in this instance it doesn't at all. Therefore, somewhat predictably, all rewards, population weights, biases and rewards remain constant. We could speak of a forceful elimination of the cumulative bias effect, albeit not of the existing initial underlying biases, which are maintained in the form of unequal, unchanged, reward distributions across the eight demographics.

Approval rate bias constraint: The constraint, even when being given a great weight on the selection process, fails terribly and results in rampant bias and rewards, comparable only to not applying any constraint at all, as seen in Figure 3.13. All bias metrics increase over the 100 epochs, except *Presence bias*. This isn't reassuring, because in this case it only means that one single majority group is accordingly hoarding almost all selections. This makes evident how these metrics work best when interpreted globally and with the adequate context, even if this notion advises against the single-constraint methods we have tested and examined here. Initial advantageous reward and population distributions give the first group an edge that is maintained and expanded all throughout to tragic effect. This, in fact, stems from an unnoticed conceptual shortcoming on our metric's very definition, in Formula 4.5. By trying to minimize the absolute difference between approval rates and their collective *average* at each moment, what is being made constant is not their shared advancement, but the gap that already exists between the first group and the rest. Minimizing the absolute difference between approval rates and their *maximum*, instead, does solve this, and in that event this constraint would actually behave very similarly to the Presence bias one, although with the added merit of not exploiting the technical loophole by which the environment's evolution was almost directly prevented in the previous case.

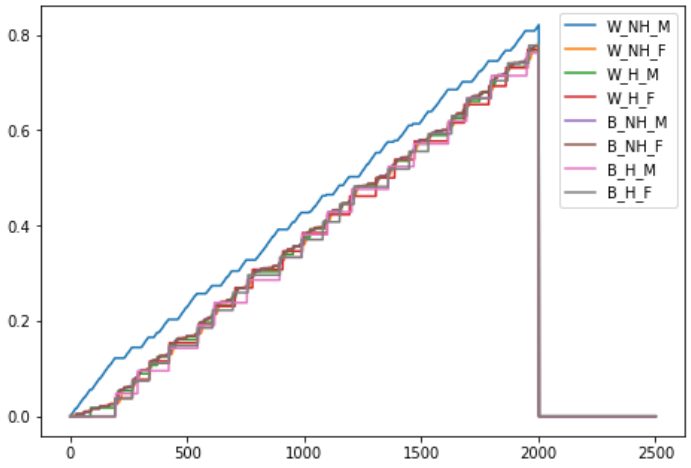

Figure 5.9: Evolution of Approval Rates within a single execution using the faulty Approval Rate constraint as-is

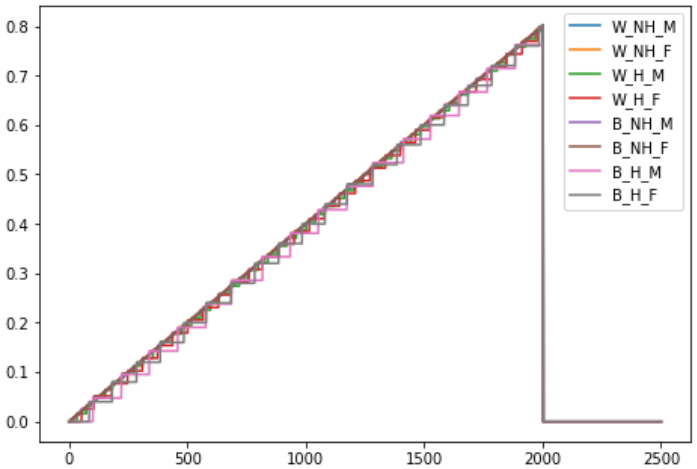

Figure 5.10: Evolution of Approval Rates within a single execution using the corrected Approval Rate constraint

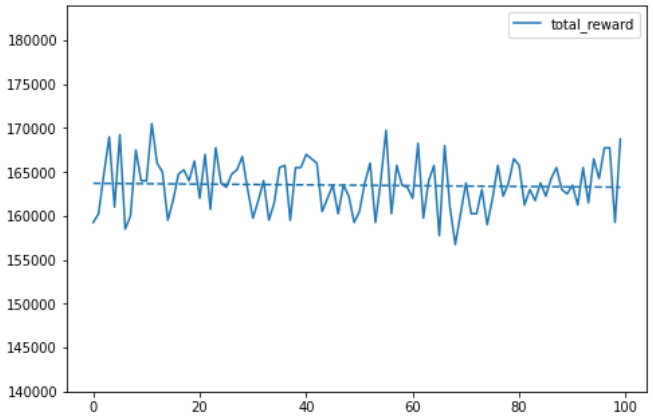

Figure 5.11: Evolution of final Total Reward for every epoch using the corrected Approval Rate constraint

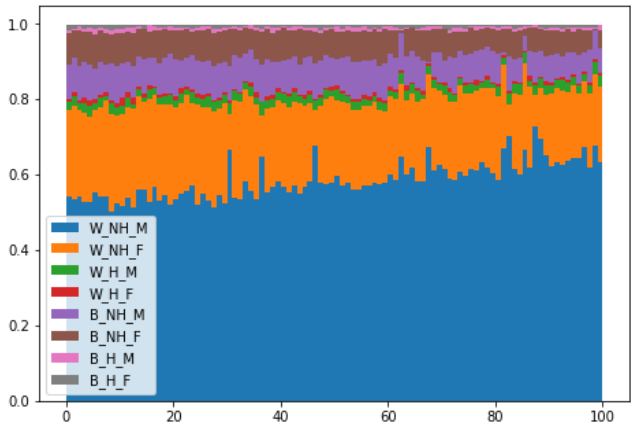

Figure 5.12: Evolution of contribution to total approvals for every epoch using the corrected Approval Rate constraint

Posteriors bias constraint: In this instance, we observe that this constraint does the best job at actually counteracting some of the initial biases that affect the environment. By the end of the 100 epochs, the proportion in which approvals are distributed is the same for all eight demographics. This experiment not only succeeds in preventing cumulative bias, but even eliminates some of the initial bias as well and we achieve that groups defined by their sensitive features do not perceive differences in their chances at being granted resources. In the final rewards Figure B.17 we perceive a period of adjustment in which rewards are lessened for a time before satisfactorily adapting and profiting from the change of paradigm in the environment. However, there are some caveats to note. Firstly, that reward distributions do not actually converge into the same degree of valuation for all demographics. This was not achieved by using any of our constraints. Secondly, in most real scenarios, we still should account for some limitations in how much the weight of a demographic or group within the data will be able to really change, increasing or decreasing, depending on a model's decisions. This is different from the level of "fluidity" of the environment, or how fast these changes will happen. This has to do with the actual population distributions present in the area the model would be used in, and how these and the ones perceived by the system can not be arbitrarily different. We did not apply this type of capping in our environment.

Rewards bias constraint: Similarly to the faulty version of the Approval Rate bias constraint that was first tested, this proved to be an arguably plain bad influence to subject our algorithm to. Selecting the seemingly less rewarding options without much moderation causes some of the least represented groups to occupy portions of the total approval significantly higher than they should according to their contribution to the data. However, in this occasion, this creates an imbalance when the rate in which their populations grow far outpaces the rate in which their rewards also increase, which makes it so that the constrained algorithm keeps prioritizing them and gives them even more weight in the total approval. On top of this circumstance, we have the fact that the reward estimates the algorithm uses to take decisions over are, of course, never their true values, and this adds a notable degree of randomness in the algorithm's behavior, as noticeable in Figure B.24. For this reason, we can conclude that this bias metric is more useful as precisely that, and not as an enforceable constraint to take decisions over. Intuitively, wanting to affect and minimize the differences between reward estimate values we don't have control over within the algorithm, and only exist and change in the environment, doesn't work as well or as reliably as the other metrics which do deal with measurements the model is directly responsible for, as are which arms have been pulled so far, at every given moment.

After these experiments, we are thoroughly reaffirmed in our impression that a different way of computing φ, and therefore choosing θ^*, could be found and could help improve or more solidly consolidate the reasoning behind the election and usage of this suboptimality coefficient, in order to be able to better defend it and more suitably navigate the tradeoff it creates between rewards and bias.

We also found that some of our metrics were more useful than others when used as a deciding factor within our algorithm, in contrast to serving as indicative measurements alone.

Finally, and on the same line of realizing and addressing some of the shortcomings in these metrics and constraint systems, we also mentioned additional ways in which some of the assumptions behind our environment, the way its properties are characterized, and the manner and extent to what it changes, certainly affected the way in which it was modeled and the way in which it behaves, and could be revised.

6. RELATED WORK

Indeed, even the type of Multi-Armed Bandit problem we were aiming to solve, and the UCB algorithm used to base our efforts on, could have been chosen to be others.

In fact, a recent work in the MAB field aimed to address a similar instance of the problem we dealt with (Tang et al., 2020), that of Multi-Armed Bandits with delayed impact of their actions, did so by using a different technique with more in common with Lipschitz (Magureanu et al., 2014) and combinatorial (Chen et al., 2016) Bandits.

This project further differs from it in that it also examines these impacts under the spotlight and perspective of AI bias, and attributes a qualitative valoration to them, in addition to quantitative. Still, it was a clear referent in terms of further basing and consolidating the belief that MAB are indeed a useful tool to approach this kind of problems too.

Alternatively, on the data science aspect of things, a resource that also served as a referent for part of this work is the already alluded to Dualfair showcasing paper (Singh et al., 2022). However, their approach to debiasing focused mainly on the proposal of new methods applicable in the pre-processing stages of data preparation, as well as original intersectional fairness metrics, both of them more directed toward their usage by more traditional ML frameworks for mortgage lending related tasks, like Logistic Regression is (Dosalwar et al., 2021).

However, the inspiration to put some distance from more data dense methods and to approach systemic bias using simulation came from another decisive paper on gender discrimination in the workplace (Du et al., 2021). On it, an agent-based environment is used to showcase how gender bias, introduced by several “mechanisms of interpersonal discrimination”, effectively establishes a glass ceiling in the organization modeled.

Finally, extensive research pieces done on the existence, definitions and repercussions of bias in AI, were also fundamental in carrying out this project (Baeza-Yates, 2016; Schwartz et al., 2022; Silberg and Manyika, 2019; Srinivasan and Chander, 2021).

7. CONCLUSIONS

This project aimed to showcase the harmful cumulative effect AI bias can have on automated decision-making systems and their non-stationary environments, and propose some methods toward addressing it. Whilst using a Multi-Armed Bandit algorithm as our model to represent one instance of said automated decision-making systems, we have tried to present new constraints to mitigate the discrimination manifest in our case of study. Basing our efforts on a simulated environment based on some key features from a selection of real data from the HMDA database, we can summarize what was achieved in this work as the following:

- Studied and identified indications of bias in the original HMDA dataset.
- Showcased the harmful effect of those underlying biases on a live environment when a MAB model computed with a UCB algorithm is applied to it sequentially over a sustained period of time.
- Developed 4 bias metrics adequate to the scope of our simulation.
- Examined the effect of using these metrics in a new version of the UCB algorithm, BC-UCB, both in an isolated environment as well as in the same live environment that changed according to its performance.

We can corroborate that the definitions and measurements made for bias can prove very dependent on the context and the problem formulation, and have to be read and interpreted in a global way.

In trying to address the problem in batch-like manner and attempting to act fairly in each iteration of the experiment, without learning much additional information about our environment outside of what's probed by UCB, we find that our best executions are not necessarily those that reduce these bias metrics over time, but those that enforce that they are lower from a start and succeed in keeping them from escalating.

Combining this notion with the assumption of fluidity given to the environment that will perceive the algorithm's decisions, we can expect that these, in turn, will succeed in getting it progressively closer to a more fair situation than if these constraints weren't used, either by preventing cumulative bias, or even reducing originally present bias itself.

We can conceive these methods, of clustering, analysis and resolution based on sensitive variables, as inspiration or as complementary tools to better root out possible indications of bias in our data and models, and control their long-term effects over the people they impact.

However, technically-inclined solutions to AI bias alone won't always be enough or the best option against it. In a perhaps seemingly counterproductive last note, and as a final word of alert which is intended to raise awareness in the reader against trusting and waiting for technology to solve everything for them, we quote Stefan Strauß, 2021, with:

> "Against this background, it is thus generally questionable whether technical fixes can effectively contribute to ease problems that are at their core sociotechnical issues with serious ethical and societal implications. [...] A system, though, that is biased and automatically attempts to debias may become even more opaque and thus uncontrollable for humans. [...] To avoid such dilemmas, there is need for broader analytical perspectives that do not just focus on the technical issues of bias but bring together different views from multiple disciplines."

Bibliography

- Angwin, Julia, et al. “Machine Bias.” *ProPublica*, 2016, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.
- Baeza-Yates, Ricardo. “Data and Algorithmic Bias in the Web.” *Proceedings of the 8th ACM Conference on Web Science - WebSci ’16*, ACM Press, 2016.
- Bartlett, Robert, et al. “Consumer-Lending Discrimination in the FinTech Era.” *Journal of Financial Economics*, vol. 143, no. 1, 2022, pp. 30–56, doi:10.1016/j.jfineco.2021.05.047.
- Chen, Wei, et al. “Combinatorial Multi-Armed Bandit with General Reward Functions.” *ArXiv [Cs.LG]*, 2016, http://arxiv.org/abs/1610.06603.
- Cowgill, Bo, and Catherine E. Tucker. “Economics, Fairness and Algorithmic Bias.” *SSRN Electronic Journal*, 2019, doi:10.2139/ssrn.3361280.
- Dastin, Jeffrey. “Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women.” *Reuters*, 2018, https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.
- Dosalwar, Sharayu, et al. “Analysis of Loan Availability Using Machine Learning Techniques.” *International Journal of Advanced Research in Science, Communication and Technology*, 2021, pp. 15–20, doi:10.48175/ijarsct-1895.
- Du, Yuhao, et al. “Insidious Nonetheless: How Small Effects and Hierarchical Norms Create and Maintain Gender Disparities in Organizations.” *ArXiv [Cs.SI]*, 2021, http://arxiv.org/abs/2110.04196.
- Federal Financial Institutions Examination Council (FFIEC). *Home Mortgage Disclosure Act (HMDA) Dataset*. 2020, https://ffiec.cfpb.gov/data-browser/data/2020?category=states&items=AL,AR,GA,MS,LA,TN&actions_taken=1,3.
- Hevelke, Alexander, and Julian Nida-Rümelin. “Responsibility for Crashes of Autonomous Vehicles: An Ethical Analysis.” *Science and Engineering Ethics*, vol. 21, no. 3, 2015, pp. 619–630, doi:10.1007/s11948-014-9565-5.
- Ji, Shuyi, et al. “Kullback-Leibler Divergence Metric Learning.” *IEEE Transactions on Cybernetics*, vol. 52, no. 4, 2022, pp. 2047–2058, doi:10.1109/TCYB.2020.3008248.
- Lattimore, Tor, and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2020.
- Ledford, Heidi. “Millions of Black People Affected by Racial Bias in Health-Care Algorithms.” *Nature*, vol. 574, no. 7780, 2019, pp. 608–609, doi:10.1038/d41586-019-03228-6.

- Magureanu, Stefan, et al. “Lipschitz Bandits: Regret Lower Bounds and Optimal Algorithms.” *ArXiv [Cs.LG]*, 2014, http://arxiv.org/abs/1405.4758.

- Momennejad, Ida, et al. “Computational Justice: Simulating Structural Bias and Interventions.” *BioRxiv*, 2019, doi:10.1101/776211.

- O’Neil, Cathy. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Penguin Books, 2017.

- Schwartz, Reva, et al. *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence*. National Institute of Standards and Technology, 2022.

- Silberg, Jake, and James Manyika. “Notes from the AI Frontier: Tackling Bias in AI (and in Humans).” *McKinsey Global Institute*, 2019, https://www.mckinsey.com/~/media/mckinsey/featured%20insights/artificial%20intelligence/tackling%20bias%20in%20artificial%20intelligence%20and%20in%20humans/mgi-tackling-bias-in-ai-june-2019.ashx.

- Singh, Arashdeep, et al. “Developing a Novel Fair-Loan Classifier through a Multi-Sensitive Debiasing Pipeline: DualFair.” *Machine Learning and Knowledge Extraction*, vol. 4, no. 1, 2022, pp. 240–253, doi:10.3390/make4010011.

- Srinivasan, Ramya, and Ajay Chander. “Biases in AI Systems: A Survey for Practitioners.” *ACM Queue: Tomorrow’s Computing Today*, vol. 19, no. 2, 2021, pp. 45–64, doi:10.1145/3466132.3466134.

- Strauß, Stefan. “Deep Automation Bias: How to Tackle a Wicked Problem of AI?” *Big Data and Cognitive Computing*, vol. 5, no. 2, 2021, p. 18, doi:10.3390/bdcc5020018.

- Tang, Wei, et al. “Bandit Learning with Delayed Impact of Actions.” *ArXiv [Cs.LG]*, 2020, http://arxiv.org/abs/2002.10316.

- Vernade, Claire, Alexandra Carpentier, et al. “Linear Bandits with Stochastic Delayed Feedback.” *ArXiv [Stat.ML]*, 2018, http://arxiv.org/abs/1807.02089.

- Vernade, Claire, Olivier Cappé, et al. “Stochastic Bandit Models for Delayed Conversions.” *ArXiv [Cs.LG]*, 2017, http://arxiv.org/abs/1706.09186.

- Weber, Mark, et al. “Black Loans Matter: Distributionally Robust Fairness for Fighting Subgroup Discrimination.” *ArXiv [Cs.CY]*, 2020, http://arxiv.org/abs/2012.01193.

- Xia, Yingce, et al. “Thompson Sampling for Budgeted Multi-Armed Bandits.” *ArXiv [Cs.LG]*, 2015, http://arxiv.org/abs/1505.00146.

APPENDICES

Appendix A: Application in a Stationary Environment

a) Presence Bias

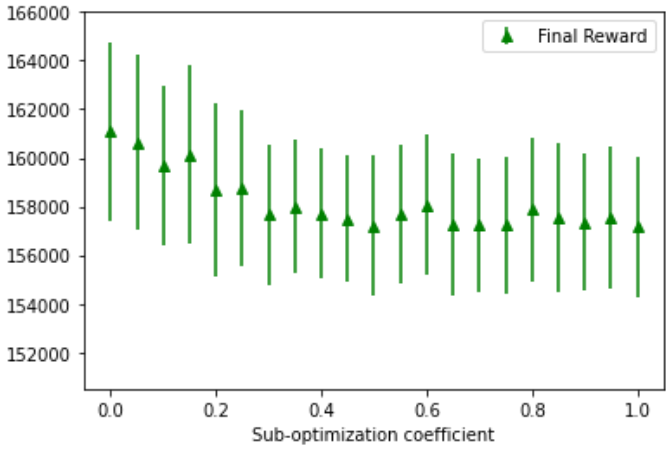

Figure A.1: Final Reward spread depending on θ when *bias_metric = bias_presence*

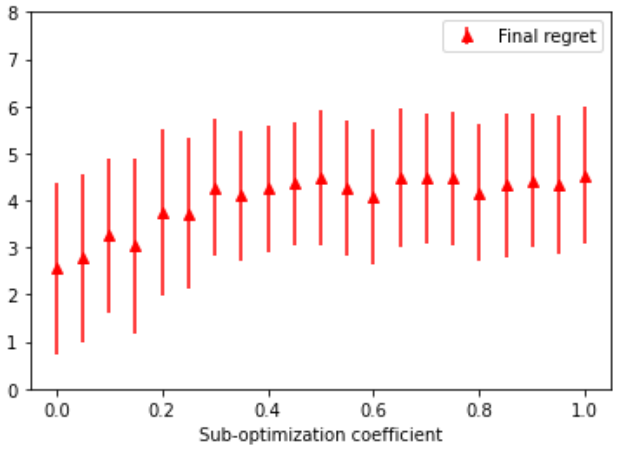

Figure A.2: Final Regret spread depending on θ when *bias_metric = bias_presence*

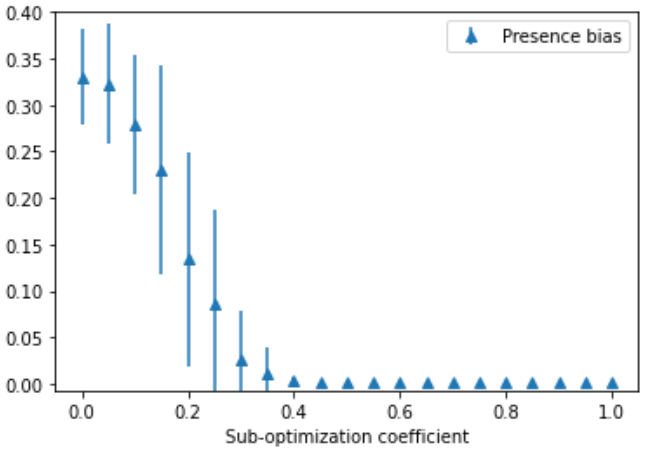

Figure A.3: Final Presence bias spread depending on θ when *bias_metric = bias_presence*

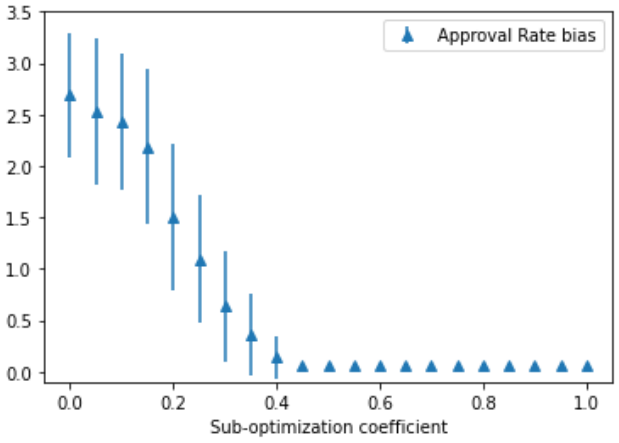

Figure A.4: Final Approval Rate bias spread depending on θ when *bias_metric = bias_presence*

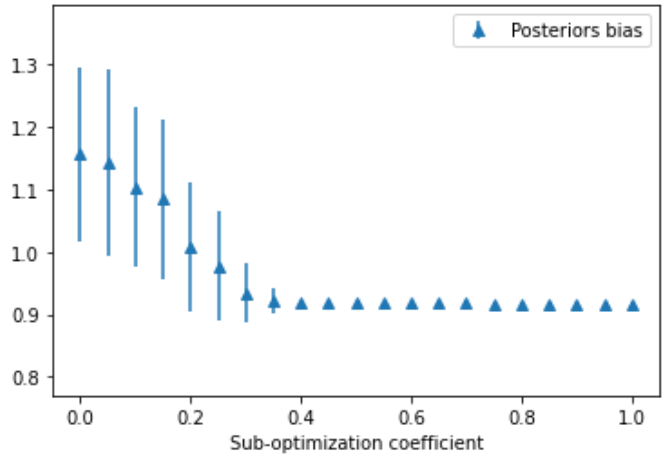

Figure A.5: Final Posteriors bias spread depending on θ when *bias_metric = bias_presence*

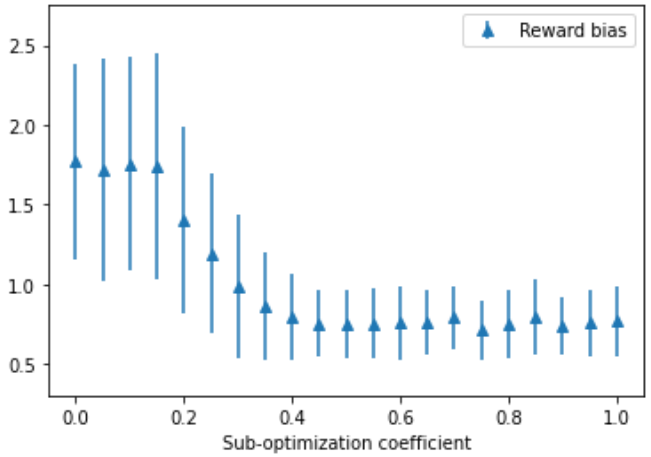

Figure A.6: Final Rewards bias spread depending on θ when *bias_metric = bias_presence*

b) Approval Rate Bias

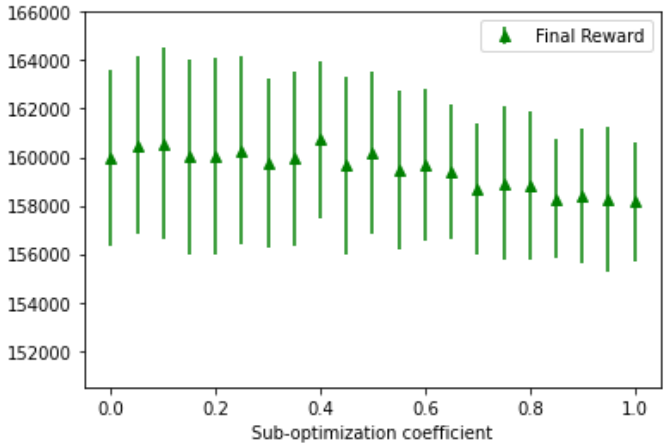

Figure A.7: Final Reward spread depending on θ when *bias_metric = bias_rates*

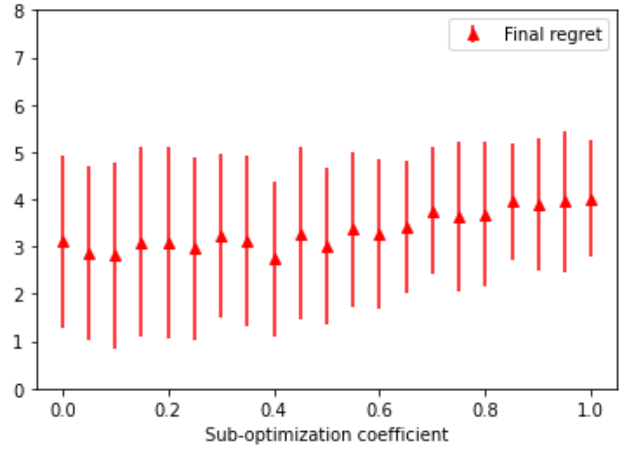

Figure A.8: Final Regret spread depending on θ when *bias_metric = bias_rates*

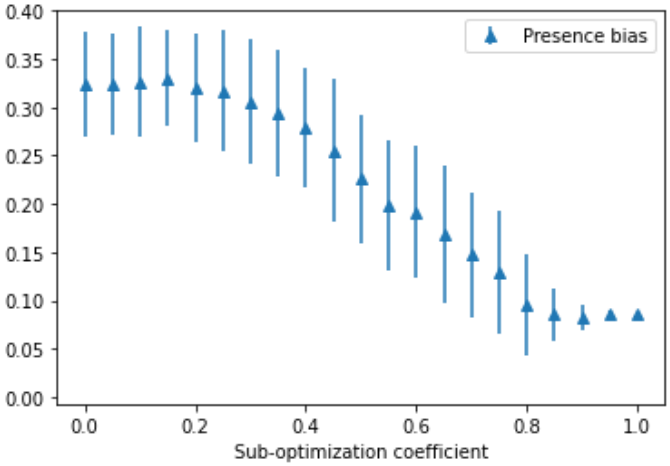

Figure A.9: Final Presence bias spread depending on θ when *bias_metric = bias_rates*

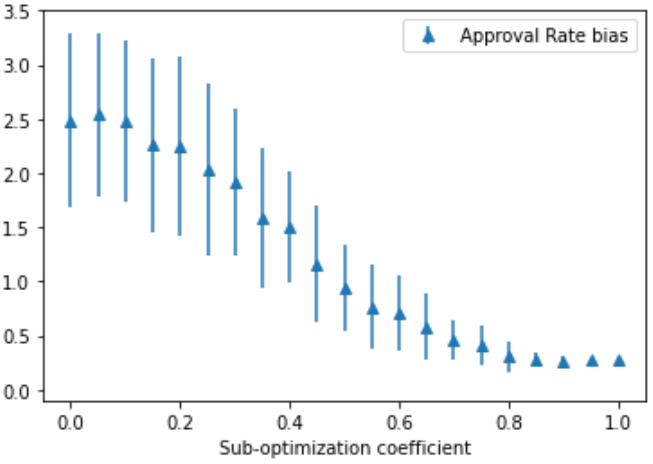

Figure A.10: Final Approval Rate bias spread depending on θ when *bias_metric = bias_rates*

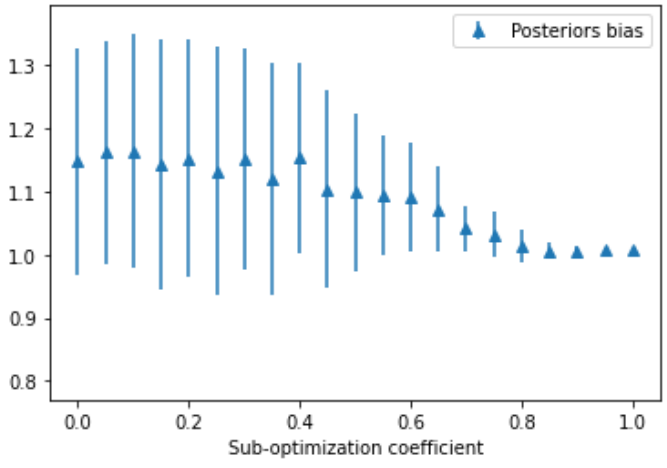

Figure A.11: Final Posteriors bias spread depending on θ when *bias_metric = bias_rates*

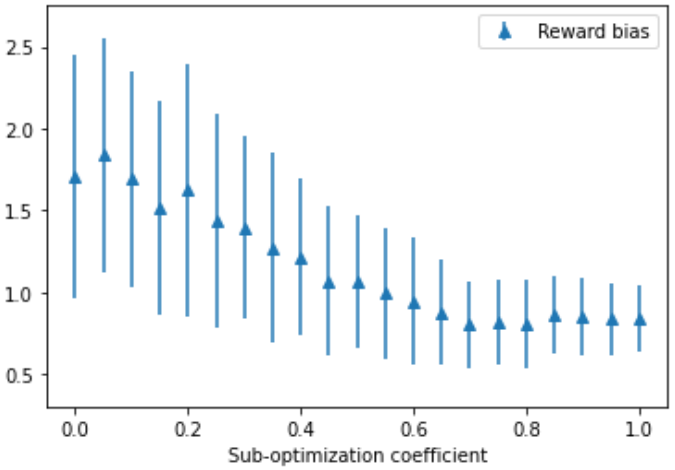

Figure A.12: Final Rewards bias spread depending on θ when *bias_metric = bias_rates*

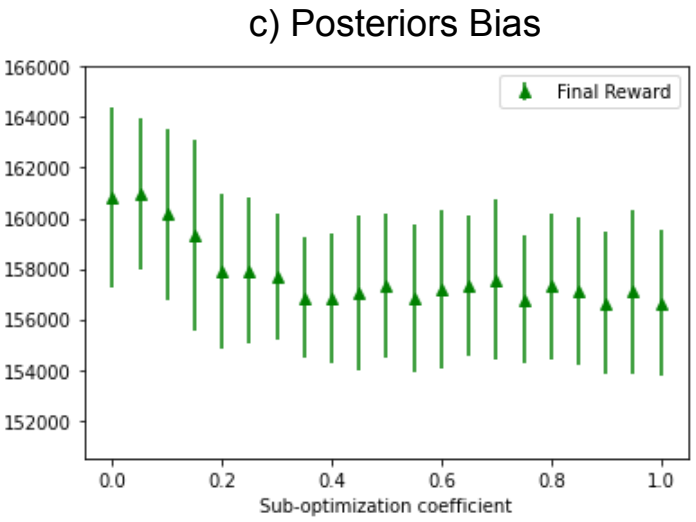

Figure A.13: Final Reward spread depending on θ when *bias_metric = bias_posteriors*

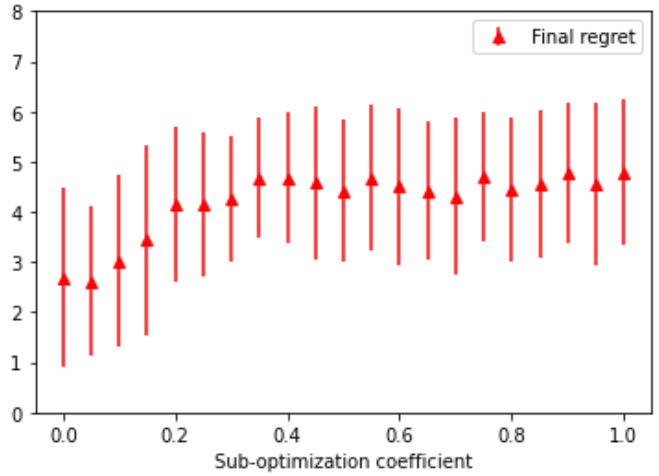

Figure A.14: Final Regret spread depending on θ when *bias_metric = bias_posteriors*

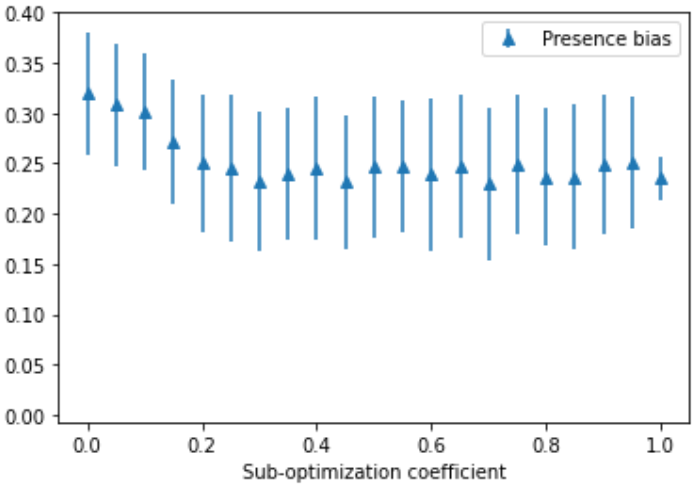

Figure A.15: Final Presence bias spread depending on θ when *bias_metric = bias_posteriors*

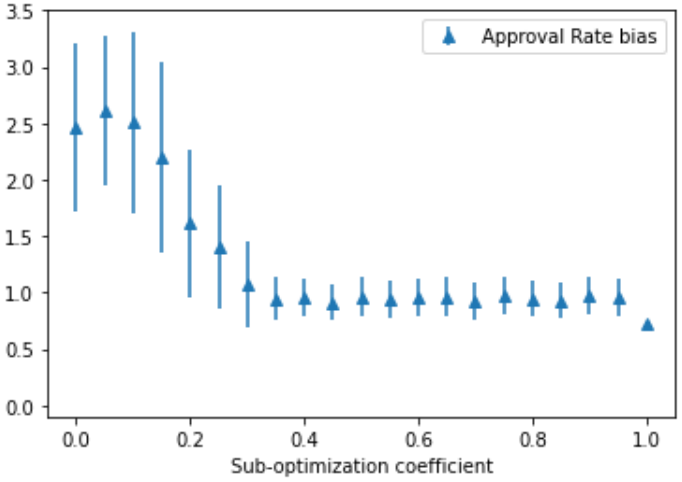

Figure A.16: Final Approval Rate bias spread depending on θ when *bias_metric = bias_posteriors*

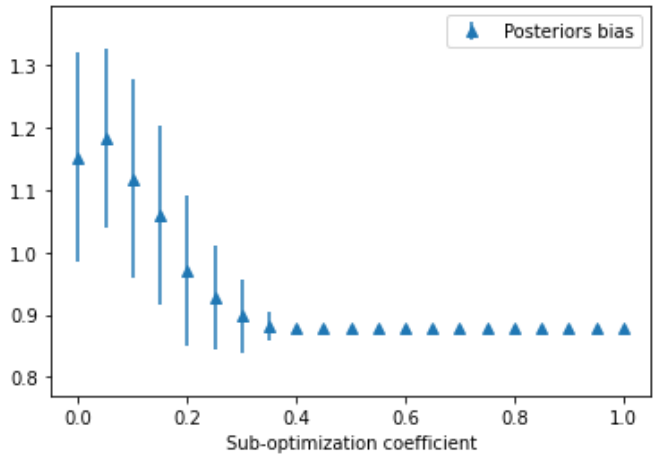

Figure A.17: Final Posteriors bias spread depending on θ when *bias_metric = bias_posteriors*

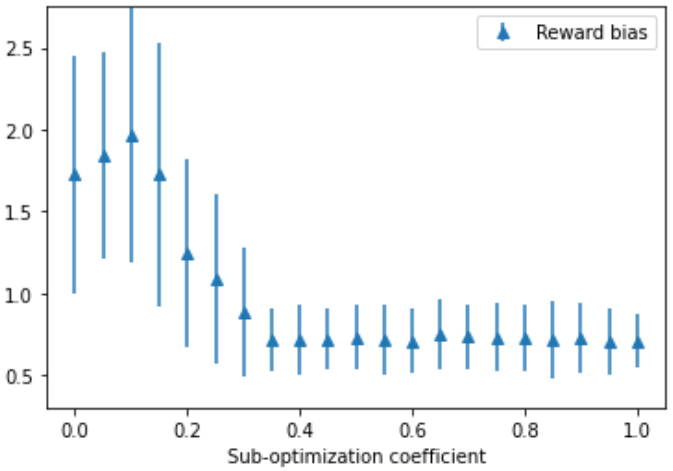

Figure A.18: Final Rewards bias spread depending on θ when *bias_metric = bias_posteriors*

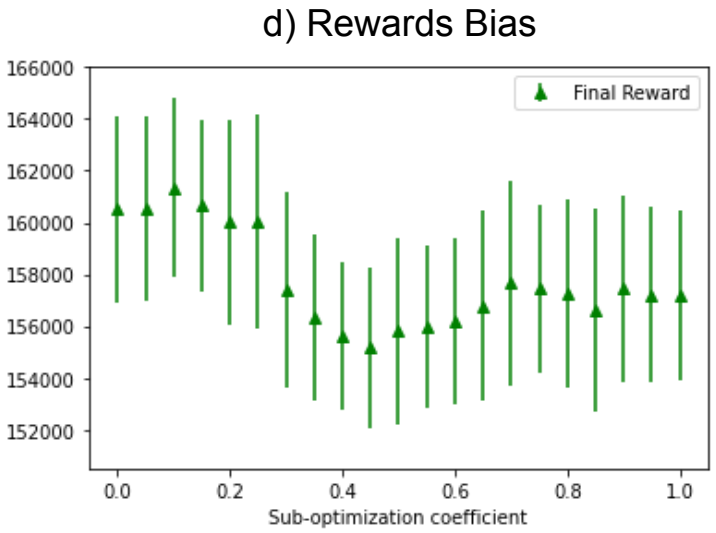

Figure A.19: Final Reward spread depending on θ when *bias_metric = bias_rewards*

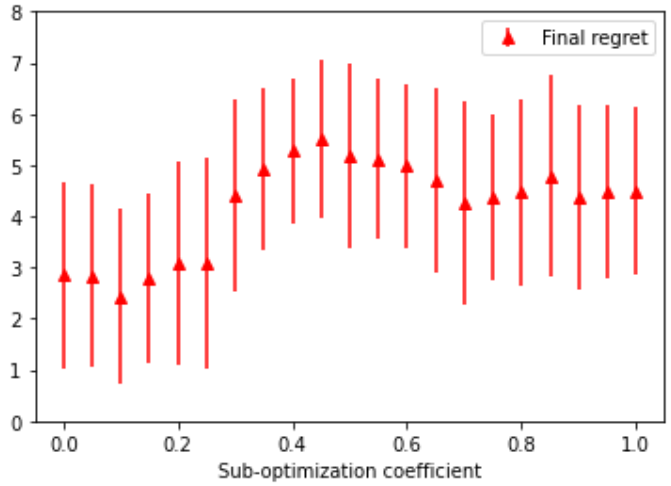

Figure A.20: Final Regret spread depending on θ when *bias_metric = bias_rewards*

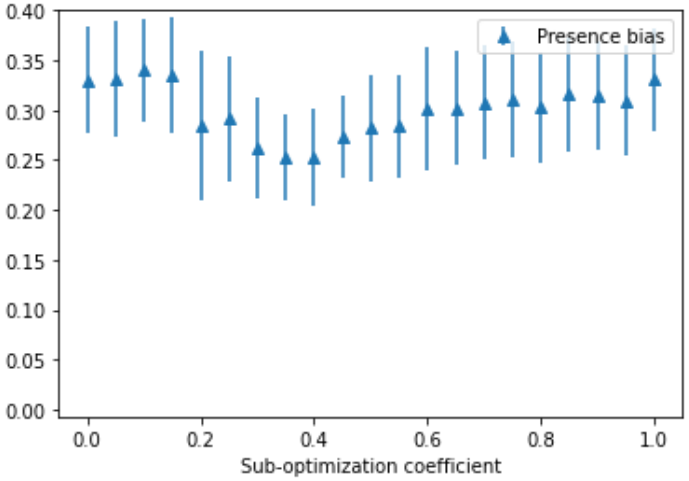

Figure A.21: Final Presence bias spread depending on θ when *bias_metric = bias_rewards*

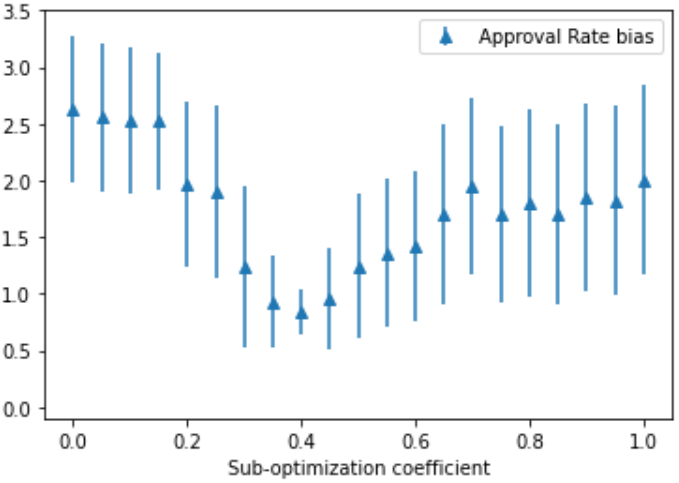

Figure A.22: Final Approval Rate bias spread depending on θ when *bias_metric = bias_rewards*

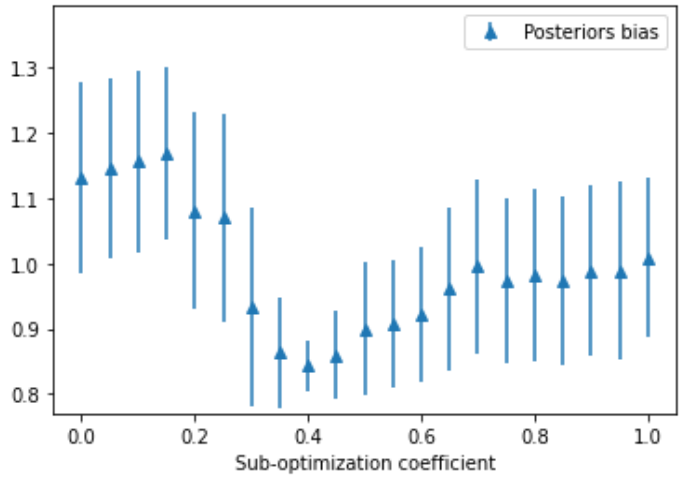

Figure A.23: Final Posteriors bias spread depending on θ when *bias_metric = bias_rewards*

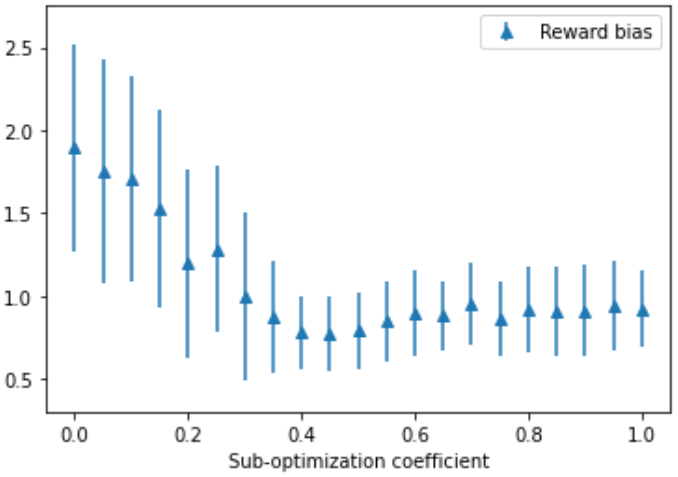

Figure A.24: Final Rewards bias spread depending on θ when *bias_metric = bias_rewards*

Appendix B: Application in a Non-Stationary Environment

a) Presence Bias (θ=0.6)

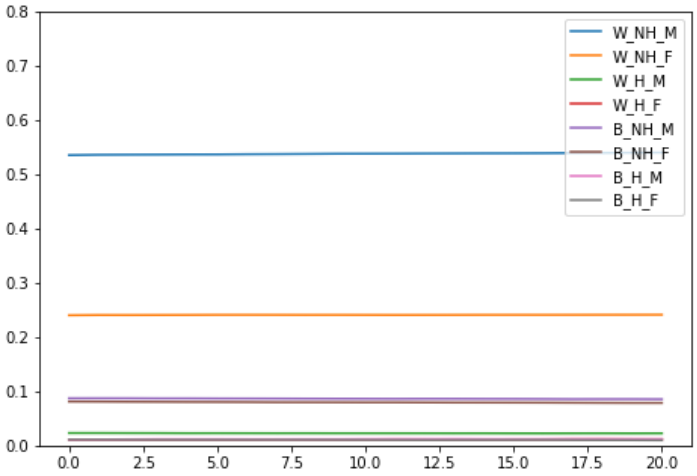

Figure B.1: Evolution of population weights every 5 epochs in a Presence-constrained environment

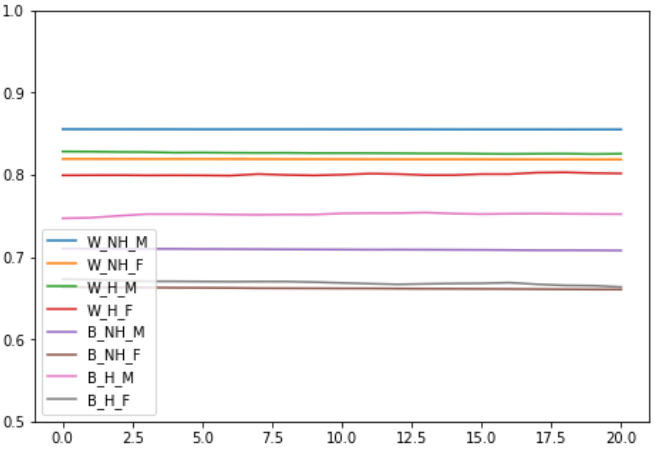

Figure B.2: Evolution of reward estimates every 5 epochs in a Presence-constrained environment

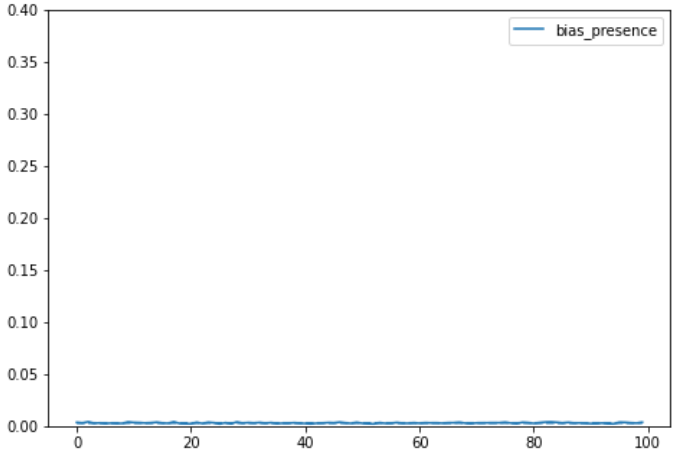

Figure B.3: Evolution of final Presence bias for every epoch in a Presence-constrained environment

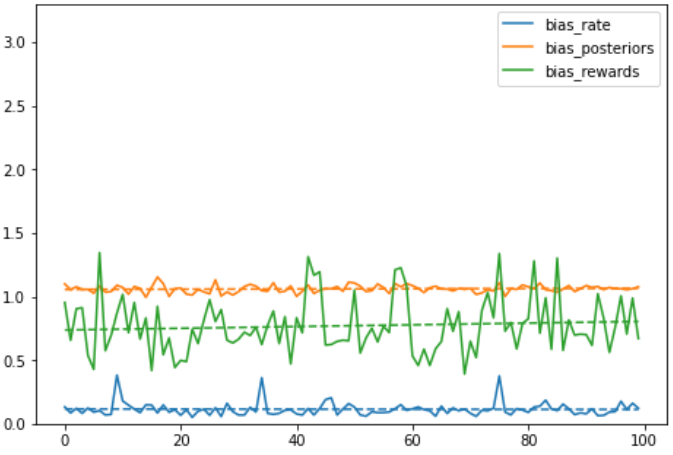

Figure B.4: Evolution of final Rate, Posteriors and Rewards biases in a Presence-constrained environment

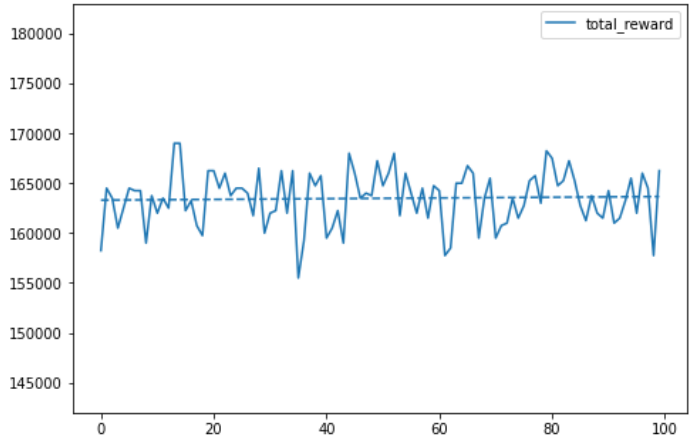

Figure B.5: Evolution of final Total Reward for every epoch in a Presence-constrained environment

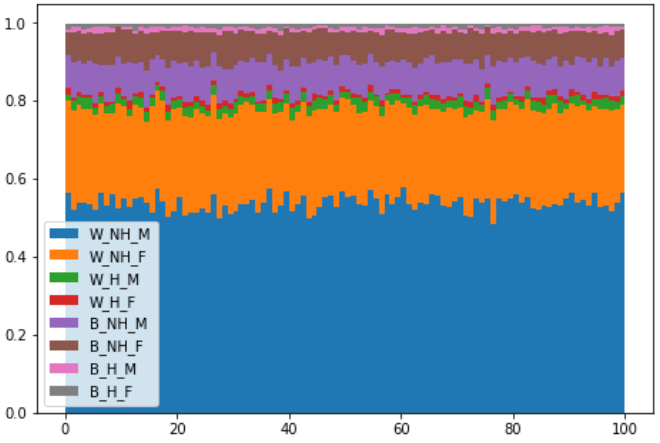

Figure B.6: Evolution of contribution to total approvals for every epoch in a Presence-constrained environment

b) Approval Rate Bias (θ=0.9)

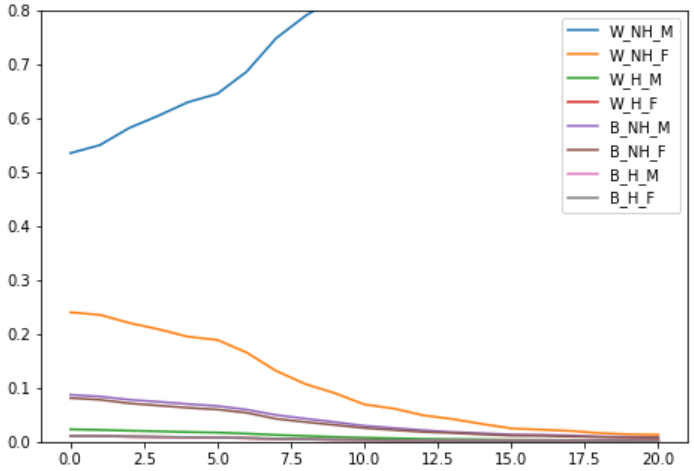

Figure B.7: Evolution of population weights every 5 epochs in an Approval Rate-constrained environment

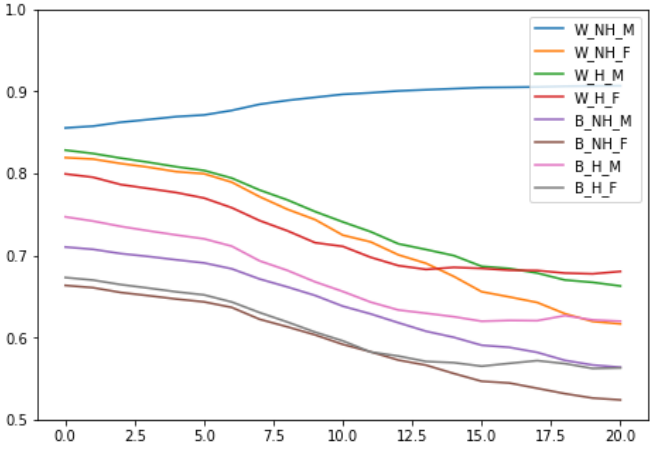

Figure B.8: Evolution of reward estimates every 5 epochs in a Approval Rate-constrained environment

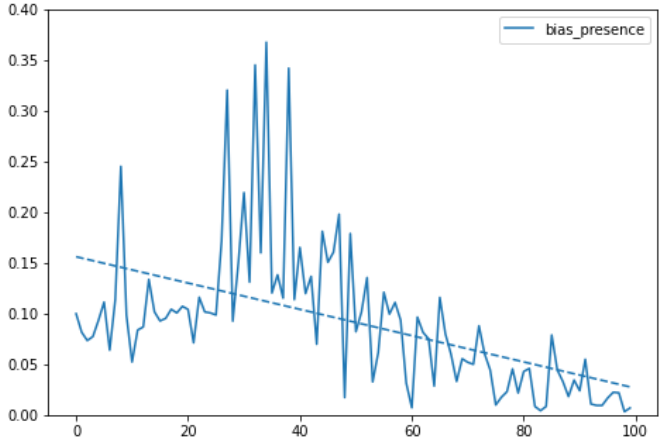

Figure B.9: Evolution of final Presence bias for every epoch in a Approval Rate-constrained environment

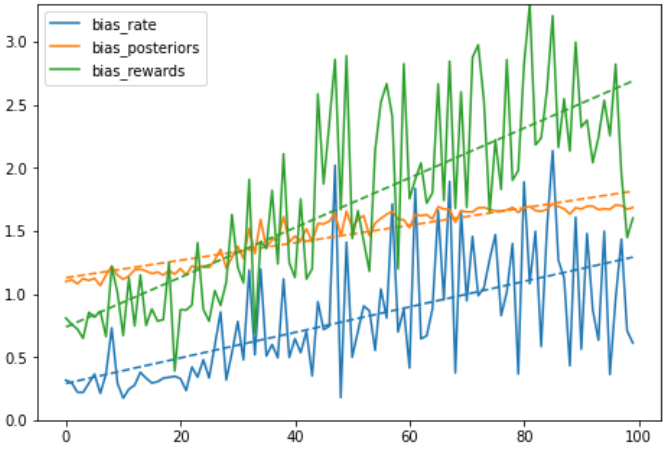

Figure B.10: Evolution of final Rate, Posteriors and Rewards biases in a Approval Rate-constrained environment

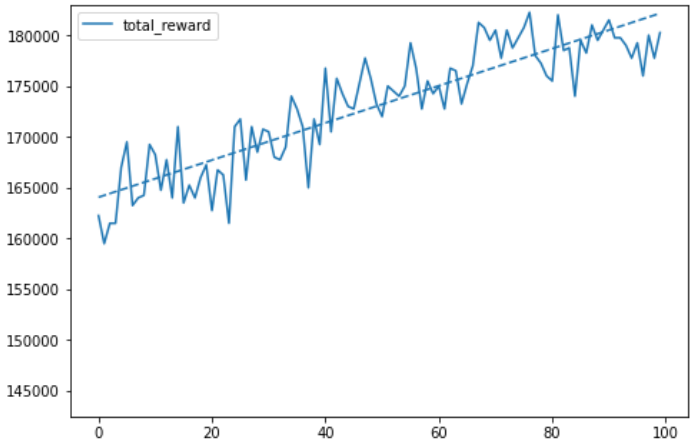

Figure B.11: Evolution of final Total Reward for every epoch in a Approval Rate-constrained environment

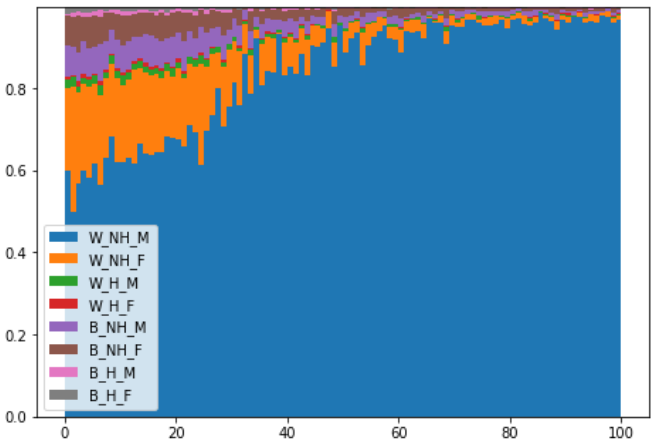

Figure B.12: Evolution of contribution to total approvals for every epoch in a Approval Rate-constrained environment

c) Posteriors Bias (θ=0.7)

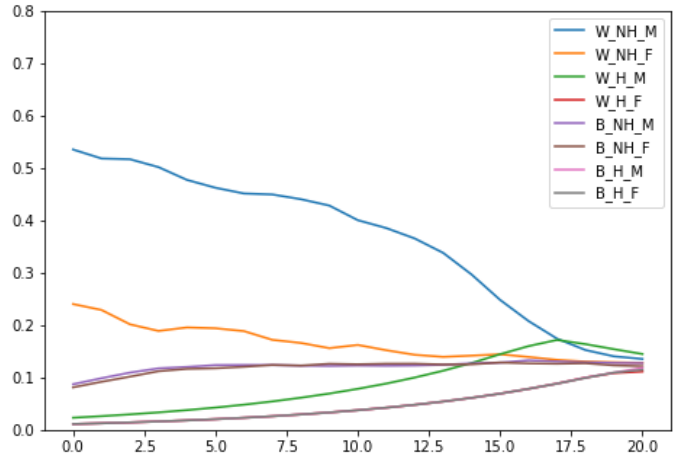

Figure B.13: Evolution of population weights every 5 epochs in a Posteriors-constrained environment

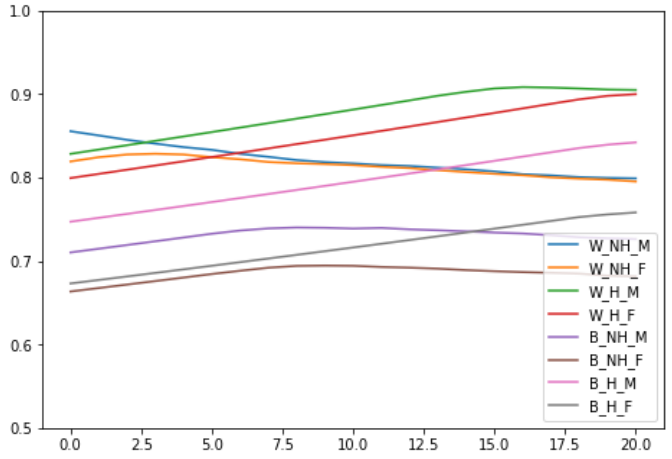

Figure B.14: Evolution of reward estimates every 5 epochs in a Posteriors-constrained environment

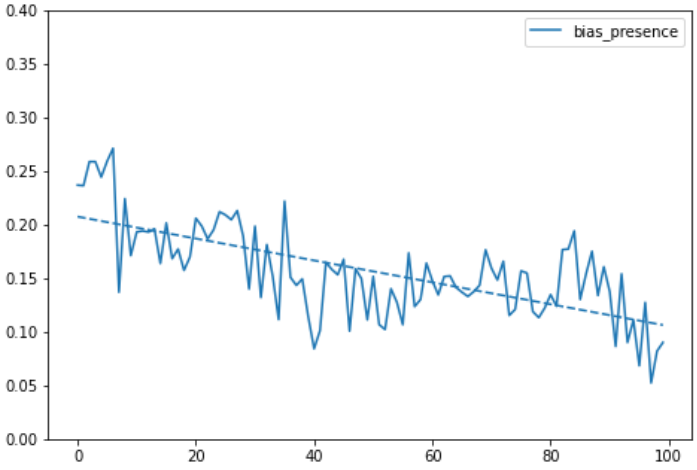

Figure B.15: Evolution of final Presence bias for every epoch in a Posteriors-constrained environment

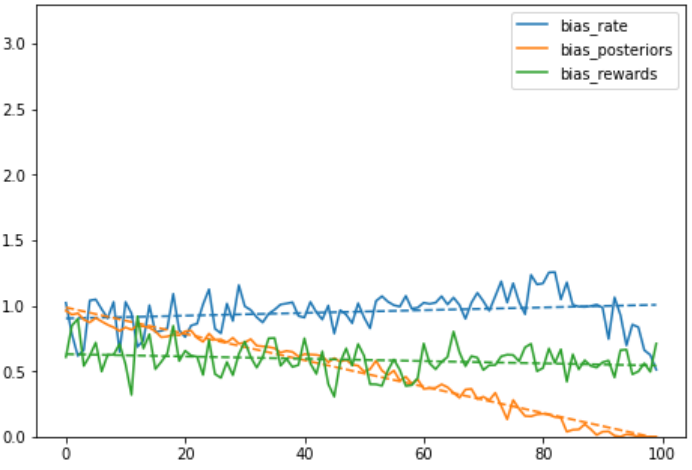

Figure B.16: Evolution of final Rate, Posteriors and Rewards biases in a Posteriors-constrained environment

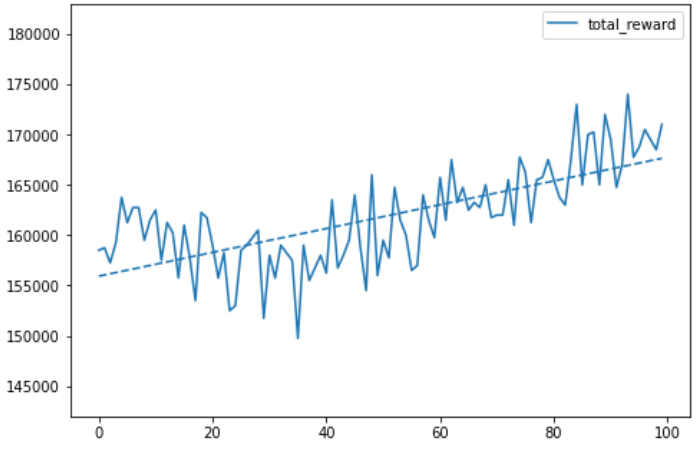

Figure B.17: Evolution of final Total Reward for every epoch in a Posteriors-constrained environment

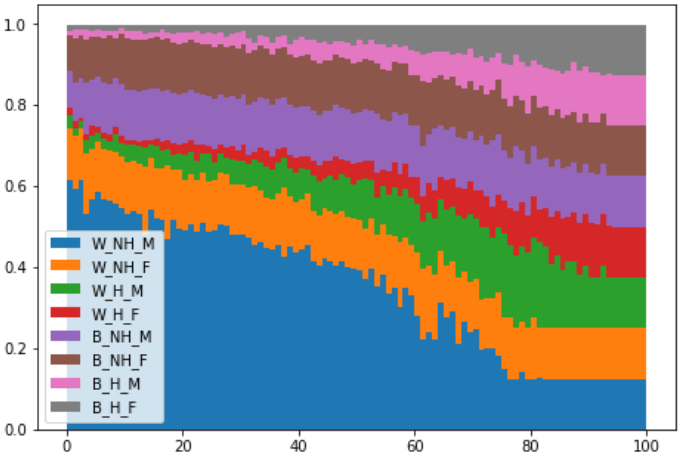

Figure B.18: Evolution of contribution to total approvals for every epoch in a Posteriors-constrained environment

d) Rewards Bias (θ=0.45)

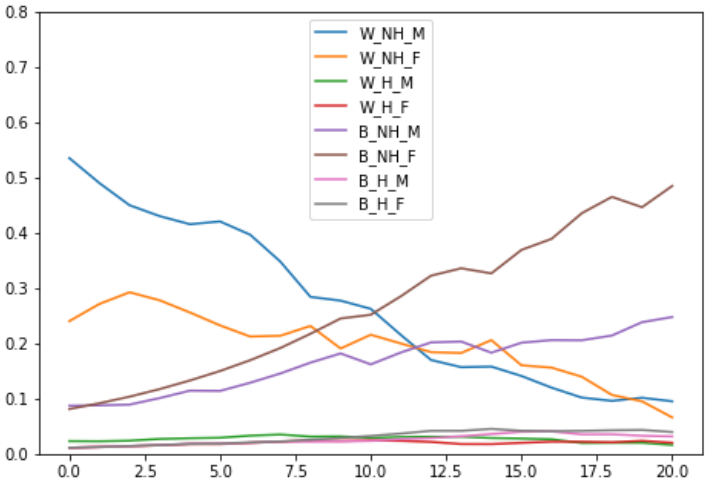

Figure B.19: Evolution of population weights every 5 epochs in a Rewards-constrained environment

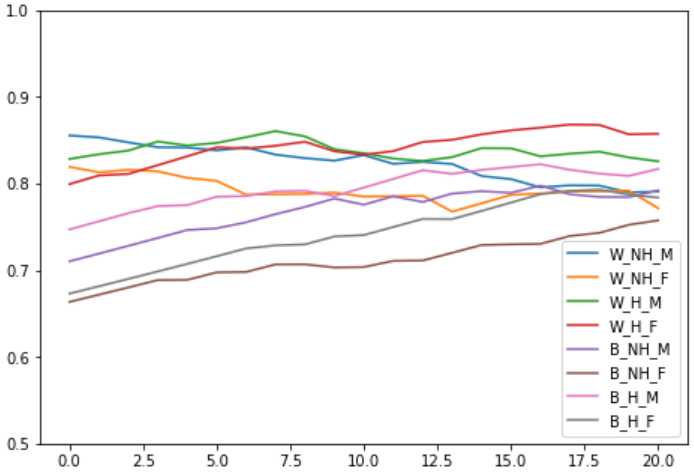

Figure B.20: Evolution of reward estimates every 5 epochs in a Rewards-constrained environment

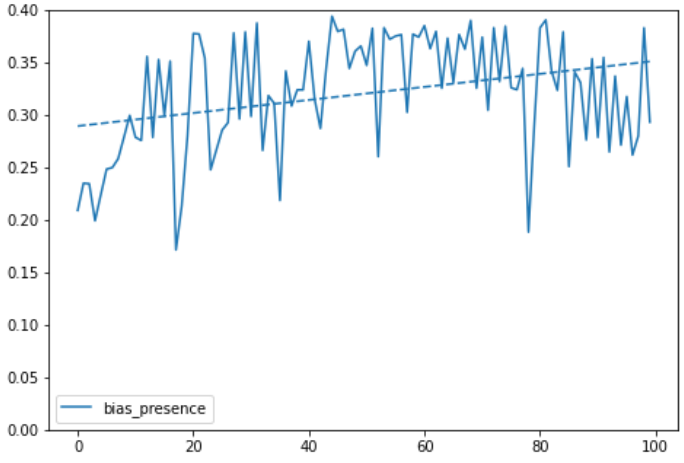

Figure B.21: Evolution of final Presence bias for every epoch in a Rewards-constrained environment

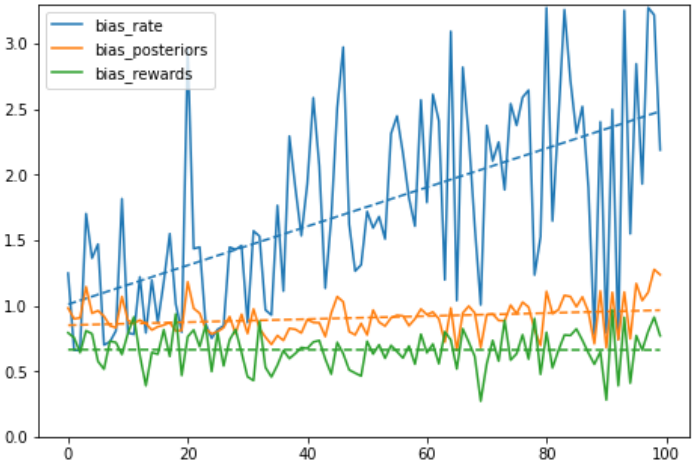

Figure B.22: Evolution of final Rate, Posteriors and Rewards biases in a Rewards-constrained environment

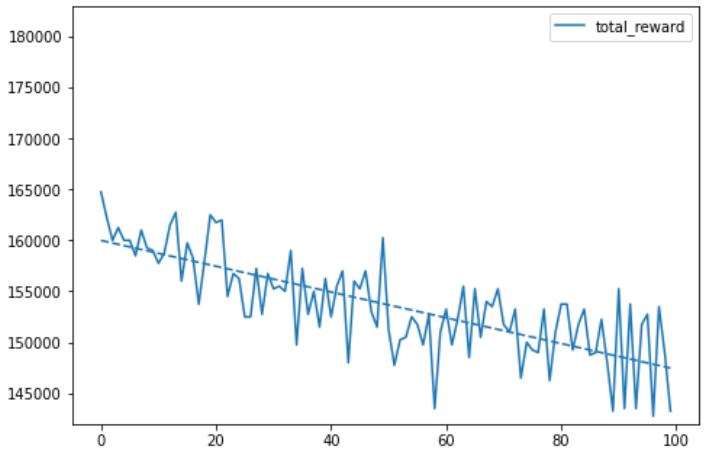

Figure B.23: Evolution of final Total Reward for every epoch in a Rewards-constrained environment

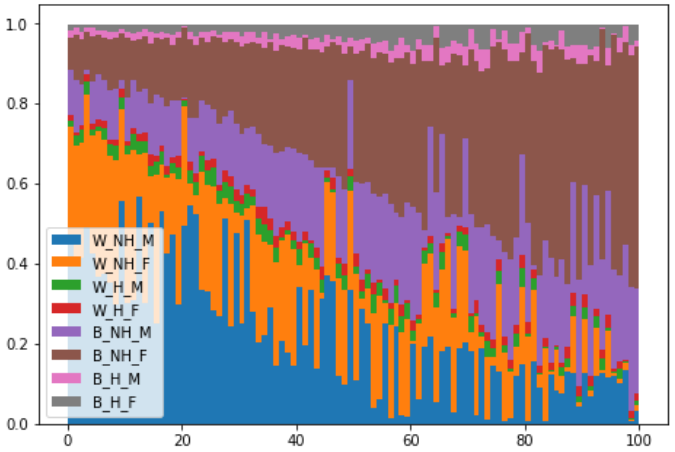

Figure B.24: Evolution of contribution to total approvals for every epoch in a Rewards-constrained environment