Improving Fairness in Sequential Decision Making through Standardized Baseline Simulations

Cassandra Ponce Maldonado, Josue Martinez,
Suprith Krishnakumar, Xicotencatl Reyes

Fall 2022

1 Research Context and Problem Statement

In recent years, Machine Learning (ML) algorithms have seen a large increase in the diversity of applications they have been employed in. Today, ML algorithms are at the forefront of many high-stake decision-making environments such as loan lending [3, 5], deployment of police officers [1], and even the healthcare industry [7]. In order to maintain impartial and just treatment across diverse communities in such high stake environments, there is an innate need to understand the possible negative effects that ML decision-makers (called agents) can have on disadvantaged or marginalized populations.

However, assessing long-term fairness of ML decision systems has remained a challenge for the ML community [4]. D'Amour et al, believe that the reason behind this is that an agent's fairness implications are mainly being tested in static environments rather than simulation environments [2]. They give an example of an agent, representing a bank, deciding to approve or deny loans from two groups with diverse credit scores. Using this environment, D'Amour et al. contrast an agent's fairness implications in static vs. simulation environments. In static environments, the analysis only entails observing if the agent was fair in giving loans to both groups of applicants, after a single round of decisions. In simulation environments however, founded upon the Markov Decision Process shown in **Figure 1**, an agent's decisions are now allowed to affect the applicants' credit scores for their next loan application. This means that the agent's decision to approve or deny a request will have significant consequences for its next iteration/ batch of decisions.

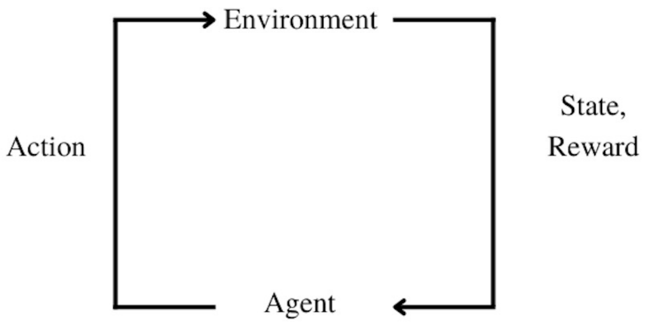

Figure 1: Outline of the Markov Decision Process. An agent takes an action that influences the state of the environment. A reward function tracks how well the agent achieved a given objective. The agent then derives feedback from the reward function and the new state of the environment, influencing the agent's next decision.

Ultimately, what D'Amour et al. found was that agents who previously performed fairly in single-step processes had significant long term fairness disparities when placed in simulations-based testing. [2].

For example, in their loan lending example, D'Amour et al. employed an *Equality of Opportunity (EO) agent* that was rewarded for minimizing the difference between the two groups' True Positive Rates. True Positive Rates were defined as an instance of a loan request being approved and successfully paid back. Thus, by attempting to minimize the disparity between the two group's True Positive Rates, the agent aimed to achieve optimal fairness performance. However, results showed that this agent overlended loans to the disadvantaged group, eventually widening the credit gap between the disadvantaged and the advantaged group [2]. Surprisingly, **figure 2** shows how this led to a less fair outcome than an agent whose purpose was to maximize bank profit without even considering equality of opportunity.

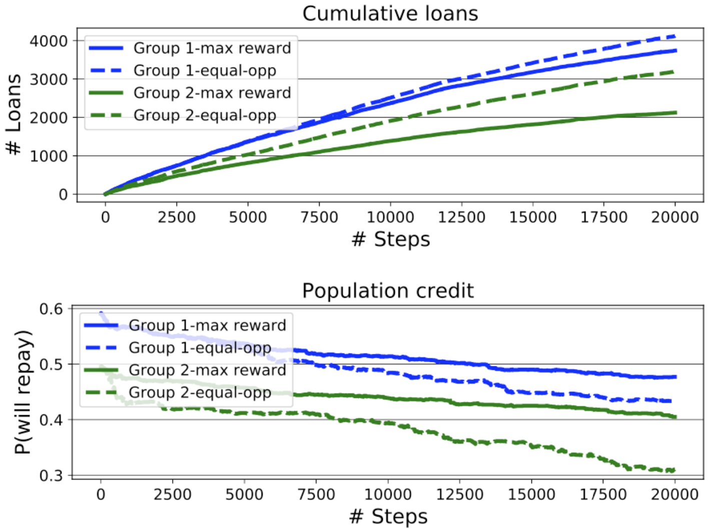

Figure 2. The second graph shows how the credit score gap between group 1 and group 2 widened over time under the *EO* agent (shown by the broken lines). Although the *EO* agent showed promise in static settings, its goal of being fair was defeated in the long-term simulation.

Evidently, the assessment of an agent's long-term fairness continues to pose a challenge in these high-stake environments since ML testing frameworks currently rely on single-step processes, instead of continuous iterations of data [2].

However, recent developments in the ML community have led to the creation of new algorithms that search for the best policy by iterating through "trial and error" in their simulation environments [4, 6, 8]. This strategy is commonly termed "policy optimization". For example, Yu et al. [9] use a fairness-constrained version of policy optimization in order to find an algorithm that maximizes the overall utility without sacrificing fairness. Using this strategy, Yu et al. were able to find algorithms that performed comparatively better than all previous algorithms in the environments from D'Amour et al.'s research [2].

Despite recent progress in developing strategies to optimize fairness, the ML community still lacks a standardized baseline to test and compare different approaches and algorithms. This means that ML researchers do not have universal simulation environments that can provide concrete metrics to assess and compare long-term fairness. Although D'Amour et al. provides their environments in an open-source library, they acknowledge that their simulations are “extremely simple” and lack sufficient complexity to provide significant fairness testing [2]. In addition, many open-source environments are difficult to adapt and deploy a decision-making algorithm on, leaving many decision systems to be without concrete simulation testing [2, 9].

Our goal is thus to provide the ML community with 3 baseline environments that are each adaptable to decision algorithms, and that provide feedback on the fairness performance of different types of decision-making systems.

2 Proposed Solution

In order to achieve this goal we will provide three environments as a baseline for testing ML researchers’ algorithms. These environments will deal with: attention allocation for incident monitoring, credit approval for lending, and disease control in population networks. These baselines will each be more flexible, easy to set up, and will include enough social context to provide substantive fairness testing [9]. Also, unlike previous work, these environments will be able to accept and interact with any ML or non-ML decision system (referred herein as agent).

2.1 Attention Allocation in Incident Monitoring

In this environment, the deployed agent will be tasked with discovering incidents across multiple sites K. The agent, however, does not have sufficient attention, N, to span all the sites, meaning that at any given moment, the agent must decide which sites to monitor for incidents. a_{kt} is the attention allocated to site k at time t, and R_{kt} is the incident rate at site k at time t. The total amount of incidents occurred at site k is $y_k t \sim \text{Poisson}(R_k t)$, and the incidents discovered is $\hat{y}_{kt} := \min(a_{kt}, y_{kt})$ [9]. All sites will also have increasing or decreasing incident rates proportional to the attention allocated to said sites by the agent: $R_{k,t+1} = R_{kt} - d * a_{kt}$ where d is the parameter that controls how dynamic the environment is and if $a_{kt} = 0$, then $R_{k,t+1} = R_{kt} + d)$ [9]. The agent will be rewarded for incident discovery and will be penalized for incidents missed using the reward function $r(st) = \zeta_0 \sum_k \hat{y}_{kt} - \zeta_1 \sum_k (y_{kt} - \hat{y}_{kt})$ which depends on ζ, a vector representing reward weights [9].

Fairness will be measured by the maximum difference between the ratios of total incidents discovered and total incidents across all sites.

$$\Delta(s_t) := \max_{k,k'} \left| \frac{\sum_t \hat{y}_{kt}}{\sum_t y_{kt}+1} - \frac{\sum_t \hat{y}_{k't}}{\sum_t y_{k't}+1} \right|$$

This will ensure that the agent does not act unfairly and over-monitor any specific site.

In terms of the fairness metrics, we will provide visual data that tracks: reward over time, true rate over time, $\Delta(s_t)$ over time, and the mean of $\Delta(s_t)$. [9].

2.2 Credit Approval for Lending

In this environment, the deployed agent will act as the decision system for a bank and will be tasked with deciding whether to accept or reject a loan request from a pool of applicants. This pool of applicants have discrete credit scores $C \in \{1, 2, ..., C_{max}\}$ and are separated into two groups, $g \in \{1, 2\}$, that are uniformly sampled [9].. We set up group 2 to be disadvantaged by having a lower initial mean credit score. The agent decides to accept or reject a loan based on the probability of the applicant repaying the loan, which is derived from a deterministic function of credit score $\eta(C)$ [9].. This decision made by the agent will affect the pool of applicants' credit scores for future time steps. In this environment, the agent will be rewarded for high bank profits defined by $r(s_t) = \zeta_0(B_{t+1} - B_t)$ where B is the bank reserve [9].. To ensure fairness between disadvantaged and advantaged groups, fairness will be defined as the max difference between two groups' True Positives Rates.

$$\Delta(s_t) = max_{g,g'} \mid TPR_{gt} - TPR_{g't} \mid \tag{1}$$

Here, TPR_{gt} refers to the true positive rate $\frac{TP}{TP+FN}$ for group g. True Positive (TP) is when the agent accepts a loan request and the loan is repaid, while False Negative (FN) is when the agent rejects a loan request that would have been repaid.

Our baseline environment will thus measure bank cash flow, cumulative loans, $\Delta(s_t)$ over time, and the mean of $\Delta(s_t)$.

2.3 Infectious Disease Control in Population Networks

For this environment, the agent is tasked with deciding which individuals, V, to vaccinate within a social network, N. N is made up of V individuals connected by E edges and the agent's goal is to minimize the spread of an infectious disease. Individuals are either susceptible (S), infected (I), or recovered (R). The infectious disease can only spread from infected individuals to susceptible individuals. Transmission will be modeled by the equation $p^v_{S \to I} = 1 - (1 - \tau)^{\#I(v,N)}$, where the probability of transmission is $\tau \in [0, 1]$ and $\#I(v, N)$ is the number of infected neighbors to individual v in N [9]. An infected individual's recovery rate is given by $p_{I \to R} = \rho$. The health of the network is thus represented by vector $H \in S, I, R^{|V|}$ [9]. The agent will have to decide to which individual to allocate a vaccine, meaning that decisions made by the said agent will innately affect the spread of the infectious disease over time. The agent is allowed to vaccinate one individual at each time step, with each individual being able to receive an unrestricted amount of vaccinations. However, vaccinations only affect individuals that are in a susceptible state and are not valid for the next time step. The agent is will be rewarded for the calculated healthiness of a population via $r(s_t) = \zeta_0 * (\sum_{i=1}^{|V|} 1(H_{it} \neq I))/|V|$ [9].

This social network will also be split into two communities by using the Girvan-Newman algorithm shown in paper [9]. Fairness will thus be described as the agent's ability to minimize the differences between the ratio of vaccinations to newly infected individuals across these two communities. Given by the following equation:

$$\Delta(s_t) = max_{c,c'} \left| \frac{\sum_t vaccinations\ given_{ct}}{\sum_t newly\ infected_{ct} + 1} - \frac{\sum_t vaccinations\ given_{c't}}{\sum_t newly\ infected_{c't} + 1} \right|$$

[9].

The baseline environment will measure an agent's reward over time, $\Delta(s_t)$ over time, and the mean of $\Delta(s_t)$.

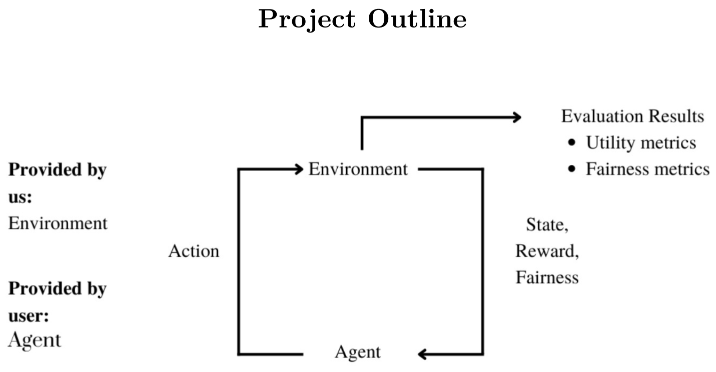

Figure 3. Outline of our project. We want to provide a standardized set of environments that have set reward functions and fairness objectives. A user will be able to test their agent by deploying it into our environments. Our focus is to provide statistics and graphs on our aforementioned metrics, reflecting the agent's performance in the simulation.

3 Evaluation and Implementation Plan

The main objective of our work is dedicated to assessing the long-term fairness of decision systems through the three specified baseline environments; environments that are likely to be used by other scientists who are interested in the long-term fairness of their decision systems. Our main evaluation method will be assessing the correctness of the fairness metrics outputted by the baseline environments across a variety of decision system algorithms. Machine Learning algorithms that don't have any fairness constraints should have more fairness violations than fairness-constrained machine learning algorithms. For our evaluation, we will be comparing the results of different agents implemented under different PPO-based policy optimizations.

We will evaluate the accuracy of our baseline by running our baseline on three established [9] agents including the Greedy PPO with no fairness constraints, Reward-Only Fairness Constrained PPO with fairness constraints at the reward level, and Advantage Regularization PPO with fairness constraints at the policy gradient level. The fairness metrics outputted by our baselines should point to ways to improve the environment dependent on the tellings of these metrics and what can be shifted to fit a more idealized form of fairness in the work of the user. This examination arrives in the form of functionality and efficiency of outputting the expected results from our designed baseline. If the baseline runs and performs well, outputting informative graphs, declarative statistics, and margins of errors dependent on the inputs from the user, then the baseline is deemed to have been executed correctly.

We believe that the purpose of these baselines will assist in maximizing fairness, equity, and inclusion for underserved communities who would otherwise be neglected due to undeveloped or outdated fairness constraints. With these baselines intact, we hope to provide the ML fairness community and by extension, society as a whole, with thoroughly tested and *fairer* ML Decision Systems.

4 Timeline

- Fall 2022
    - Week 1-2
        - Read the dedicated research paper
        - Communicating through research group etiquette
        - Met with advisors to understand the basis of our project
    - Week 3-5
        - Review Python syntax and functionality in preparation for the reinforcement learning module
        - Exposed to our advisor's published research paper
        - The draft begins for our proposal
    - Week 5-7
        - Working on Machine Learning: Reinforcement learning in preparation for code exposure.
        - Continued interaction with our advisor's research paper.
        - Evaluate new metrics for our implementation
    - Week 7-10

    - Implement metrics into code and begin construction of a new baseline.
    - Complete proposal

- Winter 2023
  - Week 1- 3
    - Plan out the structure of the Attention Allocation Environment Baseline
    - Develop Attention Allocation Environment Baseline selected decision system algorithms through Baseline, to measure their success
    - Adapt Attention Allocation Environment Baseline as needed
  - Week 4 - 7
    - Plan out the structure of the Loan Lending Environment Baseline
    - Develop Loan Lending Environment Baseline
    - Run selected decision system algorithms through Baseline, to measure their success
    - Adapt Loan Lending Environment Baseline as needed
  - Week 7 - 10
    - Plan out the structure of the Vaccine Distribution Environment Baseline
    - Develop Vaccine Distribution Environment Baseline
    - Run selected decision system algorithms through Baseline, to measure their success
    - Adapt Vaccine Distribution Environment Baseline as needed
- Spring 2023
  - Week 1- 3
    - Collect data on the success of the baselines on different decision systems algorithms
    - Begin Working on the data analysis of each Baseline's success in assessing long-term fairness
    - Debrief after data analysis and change baselines as needed
  - Week 4 - 6

    - Finalize baselines and data analysis of their success rate
    - Begin working on a poster
  - Week 7 - 8
    - Continue working on the poster
    - Create Data Visualisation figures to display baseline success
    - Create Data Visualisation figures to display fairness metrics outputted by Baseline
  - Week 9
    - Finalize Poster
    - Organize the structure of the presentation
    - Practice Presentation

5 Revisions

- Added more context to the term PPOs
- Add more information regarding the fairness constraints of each PPO-based policy optimization.
- Contextualized Markov Decision Processes in a clearer way and drew ties back to the figures we included.
- Defined unfamiliar terms and provided context for PPO in our evaluation plan that lacked depth.
- Clarified the goals of the baselines.
- Added the equations used to measure fairness for each baseline.
- Expanded on the evaluation and the range of algorithms that the baseline can be used on
- Improved grammar and sentence structure throughout the proposal
- Defined fairness in the research context section to provide a general definition of fairness

References

[1] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[2] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D. Sculley, and Yoni Halpern. Fairness is not static: Deeper understanding of long term fairness via simulation studies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 525–534, New York, NY, USA, 2020. Association for Computing Machinery.

[3] ANDREAS FUSTER, PAUL GOLDSMITH-PINKHAM, TARUN RAMADORAI, and ANSGAR WALTHER. Predictably unequal? the effects of machine learning on credit markets. *The Journal of Finance*, 77(1):5–47, 2022.

[4] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, and Yongfeng Zhang. Towards long-term fairness in recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 445–453, New York, NY, USA, 2021. Association for Computing Machinery.

[5] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA, 2016. Curran Associates Inc.

[6] Chenghao Li, Xiaoteng Ma, Li Xia, Qianchuan Zhao, and Jun Yang. Fairness control of traffic light via deep reinforcement learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 652–658, 2020.

[7] Znaonui Liang, Gang Zhang, Jimmy Xiangji Huang, and Qmming Vivian Hu. Deep learning for healthcare decision making with emrs. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 556–559, 2014.

[8] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization, 2020.

[9] Eric Yang Yu, Zhizhen Qin, Min Kyung Lee, and Sicun Gao. Policy optimization with advantage regularization for long-term fairness in decision systems, 2022.