The latest news from Google AI

ML-fairness-gym: A Tool for Exploring Long-Term Impacts of Machine Learning Systems

Wednesday, February 5, 2020

Posted by Hansa Srinivasan, Software Engineer, Google Research

Machine learning systems have been increasingly deployed to aid in high-impact decision-making, such as determining criminal sentencing, child welfare assessments, who receives medical attention and many other settings. Understanding whether such systems are fair is crucial, and requires an understanding of models' short- and long-term effects. Common methods for assessing the fairness of machine learning systems involve evaluating disparities in error metrics on static datasets for various inputs to the system. Indeed, many existing ML fairness toolkits (e.g., AIF360, fairlearn, fairness-indicators, fairness-comparison) provide tools for performing such error-metric based analysis on existing datasets. While this sort of analysis may work for systems in simple environments, there are cases (e.g., systems with active data collection or significant feedback loops) where the *context* in which the algorithm operates is critical for understanding its impact. In these cases, the fairness of algorithmic decisions ideally would be analyzed with greater consideration for the environmental and temporal context than error metric-based techniques allow.

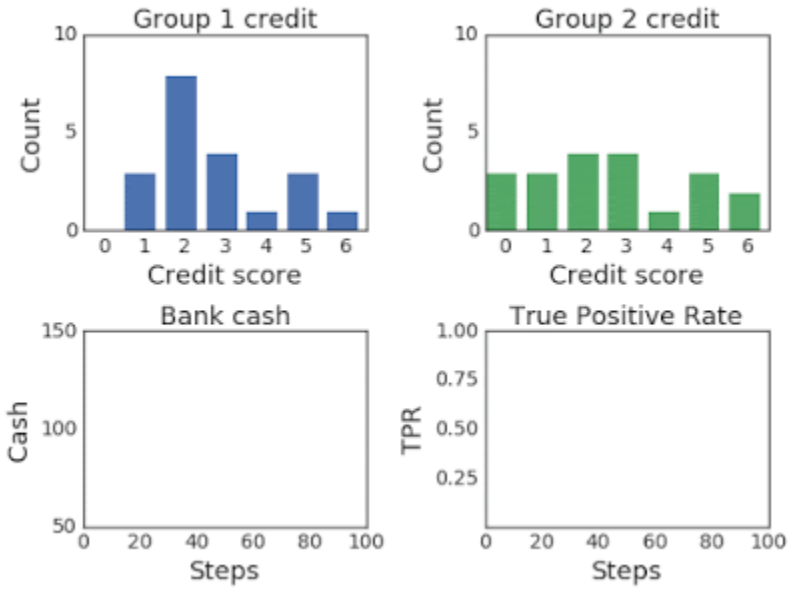
In order to facilitate algorithmic development with this broader context, we have released ML-fairness-gym, a set of components for building simple simulations that explore potential long-run impacts of deploying machine learning-based decision systems in social environments. In "Fairness is not Static: Deeper Understanding of Long Term Fairness via Simulation Studies" we demonstrate how the ML-fairness-gym can be used to research the long-term effects of automated decision systems on a number of established problems from current machine learning fairness literature.

An Example: The Lending Problem

A classic problem for considering fairness in machine learning systems is the lending problem, as described by Liu et al. This problem is a highly simplified and stylized representation of the lending process, where we focus on a single feedback loop in order to isolate its effects and study it in detail. In this problem formulation, the probability that individual applicants will pay back a loan is a function of their credit score. These applicants also belong to one of an arbitrary number of groups, with their group membership observable by the lending bank.

The groups start with different credit score distributions. The bank is trying to determine a *threshold* on the credit scores, applied across groups or tailored to each, that best enables the bank to reach its objectives. Applicants with scores higher than the threshold receive loans, and those with lower scores are rejected. When the simulation selects an individual, whether or not they will pay the loan is randomly determined based on their group's probability of payback. In this example, individuals currently applying for loans may apply for additional loans in the future and thus, by paying back their loan, both their credit score and their group's average credit score increases. Similarly, if the applicant defaults, the group's average credit score decreases.

or *sensitivity*; a measure of what fraction of applicants who *would have paid back* loans were given a loan). In this scenario, machine learning techniques are employed by the bank to determine the most effective threshold based on loans that have been distributed and their outcomes. However, since these techniques are often focused on short-term objectives, they may have unintended and unfair consequences for different groups.



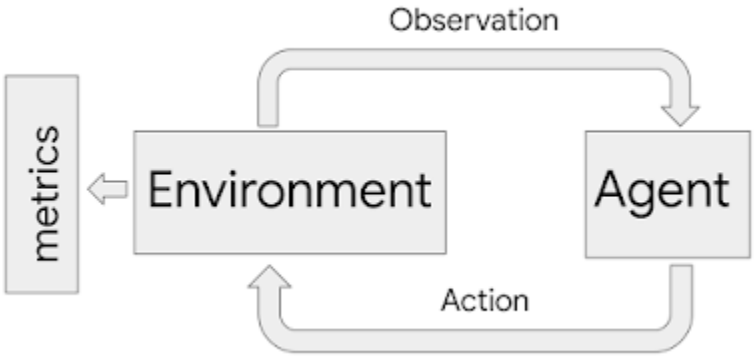
Top: Changing credit score distributions for the two groups over 100 steps of simulation. **Bottom:** (*left*) The bank cash and (*right*) the TPR for group 1 in blue and group 2 in green over the course of the simulation.

Deficiencies in Static Dataset Analysis

A standard practice in machine learning to assess the impact of a scenario like the lending problem is to reserve a portion of the data as a "test set", and use that to calculate relevant performance metrics. Fairness is then assessed by looking at how those performance metrics differ across salient groups. However, it is well understood that there are two main issues with using test sets like this in systems with feedback. If test sets are generated from existing systems, they may be incomplete or reflect the biases inherent to those systems. In the lending example, a test set could be incomplete because it may only have information on whether an applicant who has been given a loan has defaulted or repaid. Consequently, the dataset may not include individuals for whom loans have not been approved or who have not had access to loans before.

The second issue is that actions informed by the output of the ML system can have effects that may influence their future input. The thresholds determined by the ML system are used to extend loans. Whether people default or repay these loans then affects their future credit score, which then feed back into the ML system.

environment then reveals an *observation* that the agent uses to inform its subsequent actions. In this framework, environments model the system and dynamics of the problem and observations serve as data to the agent, which can be encoded as a machine learning system.

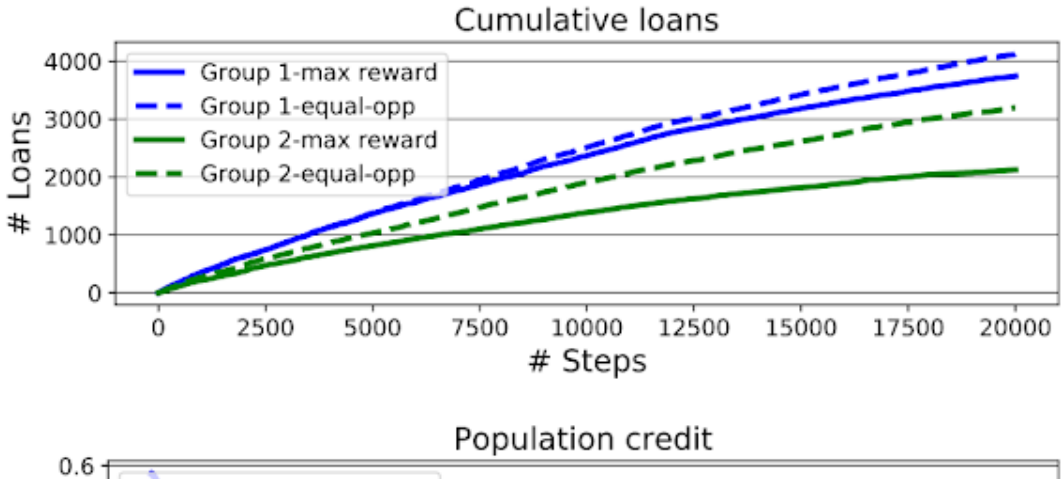


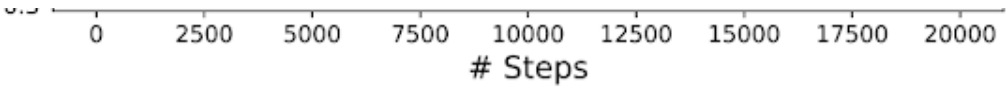
Flow chart schematic of the agent-environment interaction loop used in the simulation framework. Agents affect environments via a choice of action. Environments change in response to the action and yield parts of their internal state as an observation. Metrics examine the history of the environment to evaluate outcomes.

In the lending example, the bank acts as the agent. It receives loan applicants, their credit scores and their group membership in the form of observations from the environment, and takes actions in the form of a binary decision to either accept or reject for a loan. The environment then models whether the applicant successfully repays or defaults, and adjusts their credit score accordingly. The ML-fairness-gym simulates the outcomes so that the long-term effects of the bank's policies on fairness to the applicant population can be assessed.

Fairness Is Not Static: Extending the Analysis to the Long-Term

Since Liu et al.'s original formulation of the lending problem examined only the short-term consequences of the bank's policies — including short-term profit-maximizing policies (called the max reward agent) and policies subject to an equality of opportunity (EO) constraint — we use the ML-fairness-gym to extend the analysis to the long-term (many steps) via simulation.





Top: Cumulative loans granted by the max reward and EO agents, stratified by the group identity of the applicant. **Bottom:** Group average credit (quantified by group-conditional probability of repayment) as the simulation progresses. The EO agent increases access to loans for group 2, but also widens the credit gap between the groups.

Our long-term analysis found two results. First, as found by Liu et al., the equal opportunity agent (EO agent) overlends to the disadvantaged group (group 2, which initially has a lower average credit score) by sometimes applying a lower threshold for the group than would be applied by the max reward agent. This causes the credit scores of group 2 to decrease more than group 1, resulting in a wider credit score gap between the groups than in the simulations with the max reward agent. However, our analysis also found that while group 2 may seem worse off with the EO agent, from looking at the Cumulative loans graph, we see that the disadvantaged group 2 receives significantly more loans from the EO agent. Depending on whether the indicator of welfare is the credit score or total loans received, it could be argued that the EO agent is better or more detrimental to group 2 than the max reward agent.

The second finding is that equal opportunity constraints — enforcing equalized TPR between groups at each step — does not equalize TPR in aggregate over the simulation. This perhaps counterintuitive result can be thought of as an instance of Simpson's paradox. As seen in the chart below, equal TPR in each of two years does not imply equal TPR in aggregate. This demonstrates how the equality of opportunity metric is difficult to interpret when the underlying population is evolving, and suggests that more careful analysis is necessary to ensure that the ML system is having the desired effects.

	Group 1			Group 2			
	TP	FN	TPR	TP	FN	TPR	Eq. Opp.
Year 1	12	36	$\frac{1}{4}$	100	300	$\frac{1}{4}$	✓
Year 2	194	388	$\frac{1}{3}$	47	94	$\frac{1}{3}$	✓
Years 1+2	216	424	0.338	147	394	0.271	×