The Journal of Open Source Software

CFRL: A Python package for counterfactually fair offline reinforcement learning using data preprocessing

Several Different Contributors[1¶]

1 Several Different Departments, University of Michigan ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Reinforcement learning (RL) algorithms aim to learn a sequential decision-making rule, often referred to as a "policy", that maximizes some pre-specified benefit in an environment across multiple or even infinite time steps. It has been widely applied to fields such as healthcare, banking, and autonomous driving. Despite their usefulness, the decisions made by RL algorithms might exhibit systematic bias due to bias in the training data. For example, when using a RL algorithm to assign treatment to patients over time, the algorithm might consistently assign treatment resources to patients of some races at the expense of patients of other races. Concerns have been raised that the deployment of such biased algorithms could exacerbate the discrimination faced by socioeconomically disadvantaged groups.

To address this problem, Wang et al. (2025) extended the concept of single-stage counterfactual fairness (Kusner et al. 2017) to the multi-stage setting and proposed a data preprocessing algorithm that ensures counterfactual fairness in offline reinforcement learning. An RL policy is counterfactually fair if, at every time step, it would assign the same decisions with the same probability for an individual had the individual belong to a different subgroup defined by some sensitive attribute (such as race and gender). At its core, counterfactual fairness views the observed states and rewards as biased proxies of the (unobserved) true underlying states and rewards, where the bias is often a result of the observed sensitive attribute. In this light, the data preprocessing algorithm ensures counterfactual fairness by removing this bias from the input offline trajectories.

The `CFRL` package is built upon this definition of RL counterfactual fairness introduced in Wang et al. (2025). It implements the data preprocessing algorithm proposed by Wang et al. (2025) and provides a set of tools to evaluate the value and counterfactual fairness achieved by a given policy. In particular, it takes in an offline RL trajectory and outputs a preprocessed, bias-free trajectory, which could be passed to any off-the-shelf offline RL algorithms to learn a counterfactually fair policy. Additionally, it could also take in an RL policy and return its value and counterfactual fairness metric.

Statement of Need

Many existing Python packages implement algorithms that ensure fairness in machine learning. For example, `Fairlearn` and `aif360` focus on mitigating bias in single-stage machine learning predictions under statistical assocoiation-based fairness criterion such as demographic parity and equal opportunity (). However, they do not accommodate counterfactual fairness, which defines fairness from a causal perspective, and they cannot be easily extended to the reinforcement learning setting in general. Additionally, `ml-fairness-gym` provides tools to simulate unfairness in sequential decision-making, but it neither implement algorithms that reduce unfairness nor address counterfactual fairness (). To our current knowledge, Wang et al. (2025) is the first work to study counterfactual fairness in reinforcement learning. Correspondingly, `CFRL` is also the first code package to address counterfactual fairness in the reinforcement learning setting.

The contribution of CFRL is two-fold. First, it implements a data preprocessing algorithm that removes bias from offline RL training data. At its core, for each individual in the sample, the preprocessing algorithm estimates the counterfactual states under different sensitive attribute values and concatenates all of the individual's counterfactual states into a new state variable. The preprocessed data can then be directly used by existing RL algorithms for policy learning, and the learned policy should be approximately counterfactually fair. Second, it provides a platform for evaluating RL policies based on counterfactual fairness. After passing in a policy and a trajectory dataset from the target environment, users can assess how well the policy performs in the target environment in terms of the discounted cumulative reward and a counterfactual fairness metric. This not only allows stakeholders to test their fair RL policies before deployment but also offers RL researchers a hands-on tool to evaluate newly developed counterfactually fair RL algorithms.

High-level Design

The `CFRL` package is composed of 5 major modules. The functionalities of the modules are summarized in the table below.

Module	Functionalities
`reader`	Implements functions that read tabular trajectory data from either a `.csv` file or a `pandas.Dataframe` into a format required by `CFRL`. Also implements functions that export trajectory data to either a `.csv` file or a `pandas.Dataframe`.
`preprocessor`	Implements the data preprocessing algorithm introduced in Wang et al. (2025).
`agents`	Implements a fitted Q-iteration (FQI) algorithm, which learns RL policies and makes decisions based on the learned policy. Users can also pass a preprocessor to the FQI; in this case, the FQI will be able to take in unpreprocessed trajectories, internally preprocess the input trajectories, and directly output counterfactually fair policies.
`environment`	Implements a synthetic environment that produces synthetic data as well as a simulated environment that simulates the transition dynamics of the environment underlying some real-world RL trajectory data. Also implements functions for sampling trajectories from the synthetic and simulated environments.
`fqe`	Implements a fitted Q-evaluation (FQE) algorithm, which can be used to evaluate the value of a policy.
`evaluation`	Implements functions that evaluate the value and counterfactual fairness of a policy. Depending on the user's needs, the evaluation can be done either in a synthetic environment or in a simulated environment.

A general CFRL workflow is as follows: First, simulate a trajectory using `environment` or read in a trajectory using `reader`. Then, train a preprocessor using `preprocessor` to remove the bias in the trajectory data. After that, pass the preprocessed trajectory into the FQI algorithm in `agents` to learn a counterfactually fair policy. Finally, use functions in `evaluation` to evaluate the value and counterfactual fairness of the trained policy. See the "Example Workflows" section of the CFRL documentation for more detailed workflow examples.

Data Example

This is a data example.

Conclusions

`CFRL` is a package that empowers counterfactually fair reinforcement learning using data preprocessing. It also provides tools to evaluate the value and counterfactual fairness of a given policy. As far as we know, it is the first package to address counterfactual fairness problems in the context of reinforcement learning. Nevertheless, despite this, `CFRL` also admits a few limitations. Specifically, the current `CFRL` implementation requires every episode in the offline dataset to have the same number of time steps. Extending the package to accommodate variable-length episodes can improve its flexibility and usefulness. Besides, `CFRL` could also be made more well-rounded by integrating the preprocessor with established offline RL algorithm packages such as `d3rlpy`, or connecting the evaluation functions with popular RL environment packages such as `gym`. We leave these extensions to future updates.

Acknowledgements

This is the acknowledgements.