School of Computer Science

UNIVERSITY OF BIRMINGHAM

Intelligent Software Engineering
Lab 4: AI Model Fairness Testing

Dr. Tao Chen

This is lab4 for the module, which can be chosen as the problem for the coursework. If you need specific support in terms of programming, please attend one of the lab sessions and ask the TAs for help.

The Content

Let us recall that fairness is relevant to two concepts:

- **Sensitive feature:** The feature in the dataset that is known to be legally or ethically protected as it could influence outcomes in a way that leads to discrimination. Common examples are *Gender*, *Age*, and *Race*.
- **Non-sensitive feature:** All remaining unprotected features in the dataset would be non-sensitive features.

A generator tool is often a search algorithm, wherein the key is to pair two samples, from some testing data, that only differ on the concerned sensitive feature and randomly perturb all non-sensitive features (or sometimes any features) on both samples as guided by some fitness, hoping that such a perturbation can find more real individual discriminatory instances that would actually cause the trained AI model to predict the different outcome when changing solely on the sensitive feature. For example, suppose that the feature at index 0 is the concerned sensitive feature, then the initial sample pair $\mathbf{x}_a$ and $\mathbf{x}_b$ on the left might lead to the pair on the right after some generations:

$$\begin{aligned} \mathbf{x}_a &= \{0, 7, 4, 5, 1\} \Longrightarrow \mathbf{x'}_a = \{0, 2, 8, 5, 3\} \\ \mathbf{x}_b &= \{1, 7, 4, 5, 1\} \Longrightarrow \mathbf{x'}_b = \{1, 2, 8, 5, 3\} \end{aligned} \tag{1}$$

If both pairs can cause the trained model to generate different outcomes, then we have four individual discriminatory instances.

For Lab4, you will build a **baseline tool** using Random Sampling to automatically generate some inputs that can "trick" an AI model into making the unfair classification until the budget is exhausted, where the budget is often the maximum number of unique model inputs that can be generated. That is, our goal of fairness testing here is to find as many (unique) individual discriminatory instances as possible.

Random Search is fairly simple to implement: simply try to randomly sample n test inputs, test them over the AI model and record those that have caused it to make discrimination.

The Datasets of AI Model

This lab contains commonly used binary classification problems for fairness testing, as shown in Table **??**. These datasets are chosen because:

- They come from different domains and with a diverse number of samples and search space for perturbation.
- They contain rich characteristics and diverse demographic groups (e.g., *Age* and *Race*).

- They are real-world datasets, which strengthens the practicality of the evaluation.
- They are publicly available and are widely used in prior fairness testing studies.

All datasets come with pre-defined sensitive features. For example, the ADULT dataset has *Age*, *Race*, and *Gender*.

Table 1: The real-world datatset for fairness testing; there is a dedicated model petrained for each dataset. $|f_s|$ and $|f|$ denote the possible number of sensitive features and the number of all features, respectively.

Dataset	Domain	$\|f_s\|$	$\|f\|$	Available Size	Search Space
ADULT	Finance	3	11	$45,222$	4.81×10^9
COMPAS	Criminology	2	13	$6,172$	1.45×10^8
LAW SCHOOL	Education	2	12	$20,708$	9.20×10^6
KDD	Criminology	2	19	$284,556$	4.13×10^{15}
DUTCH	Finance	2	12	$60,420$	3.58×10^7
CREDIT	Finance	3	24	$30,000$	2.01×10^{12}
CRIME	Criminology	2	22	$2,215$	4.19×10^8
GERMAN	Finance	2	20	$1,000$	8.85×10^9

We have prepared a pre-trained AI model for each of the above datasets for you to test. The link and details of the datasets and AI model can be accessed here: https://github.com/ideas-labo/ISE/tree/main/lab4.

How to use the Model/Datasets?

The procedure would be that you load a model that is trained on a dataset, build a baseline tool, i.e., using Random Search; as part of test generation that creates different pairs of inputs for the model to predict, and record any pairs that can be termed as individual discriminatory instances and calculate the ratio as discussed below. Note that here, the randomly change inputs always come in pairs.

The Metrics

This lab is only concerned with individual fairness, which can be calculated as the ratio between the number of unique individual discriminatory instances under the true definition (I) and the size of all of the generated unique inputs (S, which is also the budget), namely IDI ratio ($\frac{I}{S}$). In fairness testing, a higher IDI ratio means more fairness bugs are found.

The Statistical Test

If there is nothing to compare, then you might not need the statistical test. The link to an existing statistical test library can be found here: `https://docs.scipy.org/doc/scipy/reference/stats.html`. You will find implementations of the tests mentioned in the module and those that we have not talked about.