EquiPy: A Python Package for Sequential Fairness using Optimal Transport with Applications in Insurance

Agathe Fernandes Machado, Arthur Charpentier, François Hu, Philipp Ratz, and Suzie Grondin

1Université du Québec à Montréal (https://fer-agathe.github.io/), 2Université du Québec à Montréal 3Université de Montréal

1. BACKGROUND: OPTIMAL TRANSPORT

For univariate distributions $p_1$ and $p_2$, the Wasserstein distance (up to 1) computes the minimum ‘cost’ required to transform $p_1$ into $p_2$:

$$W_p(ν_1, ν_2) = \left(\int_{\mathbb{R}} \left|Q_1(u) - Q_2(u)^2\right|^p du\right)^{1/p}$$

In this case, the optimal transport map $T^*$ with $T^*ν_1 = ν_2$ for some strictly convex ‘cost’, such as quadratic cost, is defined as $T^* = Q_2 ◦ T^*$. The Wasserstein Barycenter finds a representative distribution that lies between $K$ given distributions $(ν_1, . . . , ν_K)$, and weights $(w_1, . . . , w_K) ∈ R^K_+$. The $W_p^*$ Barycenter is the minimizer of

$$\text{Bar} = \left\{w_1, w_2, . . . , w_K \right\} = \arg \min_{\sum_i w_i = 1} \sum_i w_i ∥W_i - W_j∥_p$$

Hu et al. (2024) proves the associativity of the $W_p$-Barycenter with univariate measures.

2. EQUIPY: METHODOLOGY

Objective EquiPy is a Python package applying sequential fairness across a set of MSA by transforming predictions from a regression or scores from a binary classifier, denoted $f(X, A) ∈ R$, into fair ones $f_B(X, A)$. Notation

• $A = \{A_1, . . . , A_r\} = A_{r−1} \in (A_1 \times \ldots \times A_r)$, the sequence of $r$ sensitive attributes,
• $Y$ is the response variable.

Theoretical results Fair predictions, $f_B(X, A)$, are defined by:

$$f_B = \text{arg inf}_f \left\{\mathbb{E}[f(Y - f(X, A))]^2 \mid \text{risk metric} \right\}$$

$U_i(f) = \max_{ν_i} \min_{\mathbb{W}} \frac{ν_i(f_a)}{f_a}$ (unfairness regarding $A_i$)

$U(f) = \max U_i(f)$ (unfairness regarding $A$)

SSA $(r = 1)$ Given the results of Chzhen et al. (2020), $f_B$ is computed as:

$$\forall (x, a) ∈ X \times A, f_B(x, a) := \left(\sum_{a_i \in A_i} P(A_i = a_i | Q_{f(x, a)}^*) \circ F_{f_a}(x, a)\right) + 0$$

with $W_2$-Barycenter between all $f_a$ for $a_i$ in $A_i$. MSA $(r > 1)$ Hu et al. (2024) establish that $f_B$ can be expressed as:

$$\forall (x, a) ∈ X \times A, f_B(x, a) := f_{B_1} ◦ f_{B_2} ◦ \ldots ◦ f_{B_r}(x, a)$$

with the $i$-th component of a denoted $a_i$.

→ Fairness mitigation remains unaffected by the order of $A_1, \ldots, A_r$, given the property of associativity of Wasserstein barycenters.

3. LIFE INSURANCE DATA

Description The dataset, from the public SEER database, studies the mortality of US individuals with melanoma skin cancer. There are $n = 547, 878$ observations from 2004 to 2018, and 16 features describing patient characteristics (age, gender, origin) and cancer attributes (tumor size, extent).

Predictive modeling Utilizing the methodology presented in Sauce et al. (2023), we convert the dataset into survival data. Subsequently, we predict the one-year mortality while accounting for exposure over the given time interval:

1. Split the data into train and test sets,
2. Fit Logistic Regression $f$ using exposure as sample weights,
3. Apply $f$ on the test set.

→ Model-agnosticity: EquiPy employs a post-processing methodology to attain fair predictions, originating from any Machine Learning model.

4. APPLICATION IN INSURANCE

EquiPy import from equity import fairness, metrics, graphs → 3 modules

Fairness module

1. Split the test set into calibration and test sets,
2. SSA: $x_{ssa_calib}$ and $x_{ssa_test}$ contain values for a unique sensitive attribute $(A_1: \text{gender or origin})$

$wst = \text{FairWasserstein()}$

$wst.fit(scores_calib, x_{ssa_calib})$

$y_{ssa_fair} = wst.transform(scores_test, x_{ssa_test})$

3. MSA: $x_{ssa_calib}$ and $x_{ssa_test}$ contain values for both sensitive attributes $(A_1: \text{origin}, A_2: \text{gender})$

$wst = \text{MultiWasserstein()}$

$wst.fit(scores_calib, x_{ssa_calib})$

$y_{ssa_fair} = wst.transform(scores_test, x_{ssa_test})$

Metrics module unfairness(y_osa_fair, x_{ssa_test}) → $0.0478$

$y_{pred_fair} = (y_{osa_fair} > 0.102).astype(int)$

performance(y_{osa_fair}, y_{osa_test}, f1_score) → $0.1135$

Graphs module Visualization of unfairness and metrics calculations:

• fair_density_plot: representation of $f_a(x, a) ∈ A_1 \cup A_2$
• fair_multiple_actual_plot: fairness-performance relationship,
• fair_waterfall_plot: sequential gain in fairness.

References


Fair regression with Multiple sensitive attributes

Optimal Transport with Applications in Insurance

