Auditing in Interactive Machine Learning

Supervision

**Baptiste Caramiaux** (contact person, caramiaux@isir.upmc.fr)
Sorbonne Université, [https://baptistecaramiaux.com](https://baptistecaramiaux.com)

**Simone Stumpf**
University of Glasgow, [https://www.gla.ac.uk/schools/computing/staff/simonestumpf/](https://www.gla.ac.uk/schools/computing/staff/simonestumpf/)

**Téo Sanchez**
Munich Center for Digital Sciences and AI, [https://teo-sanchez.github.io/](https://teo-sanchez.github.io/)

Location and duration

The internship will be located at **ISIR** (Sorbonne University, Jussieu Campus in Paris) in the HCI Sorbonne Group.

Internship lasts 5 months, starting in March 2024

Context

Interactive Machine Teaching (IMT) systems involve people of any expertise in creating machine learning (ML) models by leveraging humans' intrinsic abilities to teach [Ramos et al, 2020]. The IMT process is iterative and engages users in a proactive process beyond data annotation. Previous work has highlighted strategies that end-users develop when placed in the role of machine teacher [Wall et al, 2019, Sanchez et al, 2020]. These strategies might result in better or worse ML models and accurate or inaccurate users' perceptions of the model behaviour [Sanchez et al, 2022].

Interactive ML and human teaching do not just consist of one phase. Instead, people spontaneously interleave teaching and testing. In teaching, the user provides training examples for the ML to learn from; in testing, the user provides test cases to see if the ML has learned the right thing. If not, further training data can be provided, and these test cases can become training examples (this is what debugging does). In short, testing is a way to audit an ML model interactively.
Problem setting

The interactive testing in IMT explores (1) how well, in general, the ML model behaves and (2) any faults (misclassifications) there are. These faults could be due to "edge cases," i.e., model blindspots [Sanchez et al, 2022]. Testing then leads to debugging, i.e., further teaching to fix any faults.

Software developers have long mastered testing. They designed robust and systematic testing routines they set up even before coding. The testing approach, which is almost a mindset, enables developers to identify and debug faults effectively. This practice can help us to study auditing in this context.

Auditing in interactive machine learning has been explored in previous work [Amershi et al. 2010; Groce et al. 2014; Chen et al. 2022]. However, much remains to be done, especially with the significant progress in explainable AI that can improve people’s perception of faults and misbehaviours. This project draws on software developers’ practices to support end-users in the process of testing in an Interactive Machine Teaching context.

Goals

1. Investigate end-user strategies for auditing in Interactive Machine Teaching
2. Develop support for IMT auditing
3. Evaluate the effects of this support in terms of (actual and perceived) fault-finding
4. Explore how to support debugging after auditing (i.e. further teaching to fix faults)

References


