Onebrain <> Workshop on Applied Machine Learning Management[1]

Description of the talk proposal

Title: “Onebrain — microprojects for data science”

Abstract

Data science at medium-to-large companies often struggles with reproducibility and reuse. Code is frequently copy/pasted instead of referenced, especially across monorepo boundaries. Onebrain is Airbnb’s solution. It invites data scientists to capture their analysis, training, evaluation, etc. code into micro-projects which abstract away CI/CD, configuration/dependency management, and command line parsing. Platforms and codebases across the company can easily reference consistent API wrappers around those projects.

Onebrain has powered a surge in the development of internal scientific libraries, creation of interactive model demos, and collaboration in the use of large language models. It has well over 200 users and over 60 distinct projects inside Airbnb after just over a year of development.

Onebrain starts with a coding standard — people write their code into an arbitrary directory structure with a standardized machine-readable (YAML) configuration file at its root. Code producers update different sections in this configuration file with meta-data related to its authorship, the software and hardware requirements for the code, and parameterized commands with which to execute it.

The main “face” of Onebrain is a command-line interface, `brain`. At a base level, this CLI facilitates and enforces the Onebrain coding standard, with commands like brain generate to initialize a new project and brain run to execute them.

On the back-end, the brain CLI is complemented by continuous integration-type jobs and server infrastructure that powers the many things that Onebrain is:

- A replicable computing environment: Onebrain project configuration files contain instructions on the type of hardware required for the project, the required operational system containerized image, and coding package dependencies that need to be installed. More importantly, Onebrain automates and hides this operational overhead from code consumers who are able to run the project with something as simple as brain run, or clicking on a URL link that the code author generated.

[1] https://wamlm-kdd.github.io/wamlm/index.html

- A template repository: when code authors publish their projects into a Onebrain code repository, any user of Onebrain is able to seamlessly invoke that project, and run it or use it as a starting point for their own work.
- A version-controlled repository and CI: publishing a Onebrain project doesn't require checking code into a centralized Onebrain repository, any code repository can be a Onebrain repository. Teams and individuals are free to use any repository they want, and their projects will all be available within the Onebrain ecosystem (one only needs to reference a centralized CI dispatch job when creating the repository).
- A code package publisher: coding package producers (e.g. R library, Python package) can write their code as a Onebrain project, and simply pushing it to a Onebrain repo will automatically publish it into a hosted package repository where anyone can install it with standard package managers or seamlessly reference it within other Onebrain projects.
- A prototype application host: Onebrain projects can accommodate application code (e.g. Streamlit, Shiny) to be hosted as a web page. Because Onebrain is already a replicable-code environment, the code can be checked into an application repo that will automatically launch the app into a hosted server. Users are then able to visit a URL and interact with the application.

Relevance to the workshop

Agility, collaboration, and lifecycle management remain constant themes in applied machine learning. Onebrain solves all three of these problems simultaneously at Airbnb.

Info

Presenter bio: Dan Miller did a PhD in number theory at Cornell, then joined Microsoft. He initially worked in the Experimentation Platform on A/B test design and understanding with Bing, then led an effort to build a new multi-target compiler ("Mangrove") for generating experiment analysis code. Then, he led an effort to build better support for privacy-preserving ML techniques like federated learning into the platform. Now, he works at Airbnb on aligning data science and machine learning to use tooling that makes collaboration and rapid iteration easy and scalable.

Company portrait: Airbnb has a data science / machine learning org of 300 people working in 5+ monorepos, building a range of models from simple decision trees for churn prediction, all the way to state-of-the-art pretrained transformer models. The company grapples with the unique challenges of a two-sided marketplace, as well as a mix of legacy and brand-new technology.