Datalab: A Version Data Management And Analytics System

Yang Zhang, Fangzhou Xu, Erwin Frise, Siqi Wu, Bin Yu, Wei Xu
Overview

- Problem: how do we manage code and data with versions?
  - Code version control, e.g. GitHub
  - Data version control, e.g. DataHub[1]
- But how to combine them in a coherent system?

Our Solution

- Version control combining codes and datasets.
- Datasets are generated by execution of codes.
- Two data versions are connected by a code version.

```
Dataset version0001
Experiment
commit_id = c41f29
Dataset version0002
```
Pairs of data versions make up a data work flow (DWF)

Reconstruct a dataset by re-executing the version of code that generates it
System Architecture
Case Study -- A Biological Data Application

- **Goal:** find the best $K$ principle patterns
- **Procedure:**
  - Data preprocessing
  - Feature extraction
  - Non-negative matrix factorization
  - Evaluate $K$ by a stability function
  - Repeat until find the best parameter

$$\text{stab}(K) = \frac{2}{B(B-1)} \sum_{i<j} \text{diss}(D_i, D_j)$$
## Core APIs

<table>
<thead>
<tr>
<th>Name</th>
<th>Functionality</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>create a project</td>
<td>dataset name</td>
<td>null</td>
</tr>
<tr>
<td>inspect</td>
<td>list metadata of a dataset</td>
<td>dataset name</td>
<td>spreadsheet</td>
</tr>
<tr>
<td>upload</td>
<td>upload a dataset to a project</td>
<td>file or directory name</td>
<td>null</td>
</tr>
<tr>
<td>import</td>
<td>import a dataset to the system</td>
<td>file or directory name</td>
<td>null</td>
</tr>
<tr>
<td>merge</td>
<td>merge two dataset into one</td>
<td>two dataset names</td>
<td>null</td>
</tr>
<tr>
<td>diff</td>
<td>check out the difference between two datasets</td>
<td>two dataset names</td>
<td>spreadsheet of difference</td>
</tr>
<tr>
<td>submit</td>
<td>push codes to system and automatically execute</td>
<td>commit ID</td>
<td>null</td>
</tr>
</tbody>
</table>
Future Work

- Dataset caching
- Online development environment
- Multi-level of interfaces
Conclusions

- We combine data and code version control
- We propose data work flow
- We improve the efficiency of a data science procedure