TIME TRAVEL AND PROVENANCE FOR MACHINE LEARNING PIPELINES

Alexandru A. Ormenisan¹,², Moritz Meister², Fabio Buso², Robin Andersson², Seif Haridi¹,², Jim Dowling¹,²
What is provenance - **ML Pipeline**

- **ML Pipeline**
  - **Feature engineering**
    - Raw Data
  - Training
    - Features
  - Serving
    - Models
What is provenance - Metadata

Feature engineering
- Raw Data
- Features
- Models
- <fg_eng, raw_data, features>

Training

Serving

Pipeline code

```python
In[]:
add(fg_eng, raw_data, features)
...add(training, features, model)
```
Implicit and Explicit Provenance

**Implicit**
- Bottom-up tracking of provenance.
- Requires changing the platform.
- Conventions link files to artifacts.
- Metadata is strongly consistent with storage platform.

**Explicit**
- Top-down tracking of provenance.
- Requires changes to application or library code.
- Metadata Store is decoupled from the data storage.

Diagram:
- Application
- Libraries
- Metadata store
- Running platform
Why track provenance?
Provenance queries

- Type
- Throughput
- Latency
- Size
Systems Challenges - Consistency

- Metadata
- ML Artifacts
- FS

Consistency issues
Synchronization

?
Let the platform manage the metadata!

- Feature engineering
- Training
- Serving

Raw Data → Features → Models

Distributed File System (HopsFS)
Change Data Capture (ePipe)
Provenance
Sql, Graph, Timeseries, **Full Text Search**
ML Artifacts

- Features, Feature Metadata
- Train/Test Datasets
- Models, Model Metadata

Possibly thousands of files

Distributed File System

Generate thousands of operations

Change Data Capture (CDC)

Capture only relevant operations
More context to base file system operations?

Are any of these operations related?

Certificates (with AppId) enabled FS Operation
Richer provenance information

Distributed File System
Read/Write/Create/Delete/XAttr/Metadata

Resource Manager - Yarn (Application Context)
Job Manager - Hopsworks (Job Context)
Workflow Manager - Airflow (Pipeline Context)

Additional Context

<Link input/output files via Apps
Different Executions of the same Job
Jobs as Stages of the same Pipeline

<file, op, user_id, app_id, job_id, pipeline_id>
Hopworks Conventions

/featurestore
  /training_datasets
  /models
  /logs
  /notebooks
CDC API - Filtering Mechanisms

- Path based filtering

Example
Project
/featurestore
/training_datasets
/models
Path based filtering

• Tag based filtering

Example:
Custom metadata based on HDFS XAttr.
Tag: <tutorial>, <debug>
Tags can enable logging of all operations, if path based filtering is not easy to set
Path based filtering

Tag based filtering

- **Coalesce FS Operations**

Example:
Read file$_1$
Read file$_2$
...
Read file$_n$

Access$_1$
Training Dataset
CDC API - Filtering Mechanisms

Path based filtering
Tag based filtering
Coalesce FS Operations

- **Filtered Operations**

<table>
<thead>
<tr>
<th>Create/Delete</th>
<th>Artifact existence</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAttr</td>
<td>Linking metadata to artifact</td>
</tr>
<tr>
<td>Read</td>
<td>Artifact usage</td>
</tr>
<tr>
<td>Children Files Create/Delete</td>
<td>Artifact mutation</td>
</tr>
<tr>
<td>Append/Truncate</td>
<td>Artifact mutation</td>
</tr>
<tr>
<td>Permissions/ACL</td>
<td>Artifact Metadata mutation</td>
</tr>
</tbody>
</table>
Hopsworks ML Pipelines

Metadata Store

Feature Engineering CI/CD Platform

Feature Store

- commit-0097
- commit-0002
- commit-0001

Model Training CI/CD Platform

Model Repository Model Serving & Monitoring

Model Training & Model Validation

1. Data
2. Feature (Data) Pipelines
3. Develop Model
4. Train/Validate Model
5. Deploy/Monitor
Provenance example

Bias Detected!

What do I do?
Provenance + Time travel

Claim of Model Bias!
Can we determine the exact features used?

- Feature engineering
- Training
- Serving
- Application
- Raw Data
- Features
- Models

Hudi Feature Timeline

- Delta Commit: 10/01/20@10:10:01
- Delta Commit: 10/01/20@10:12:01
- Delta Commit: 10/01/20@12:10:01
● Provenance improves understanding of complex ML Pipelines.

● Provenance should not change the core ML pipeline code.

● Provenance facilitates Debugging, Analyzing, Automating and Cleaning of ML Pipelines.

● Provenance and Time Travel facilitate reproducibility of experiments.
Thank You!

HOPSWORKS.ai
BY LOGICAL CLOCKS

@logicalclocks

http://github.com/logicalclocks/hopsworks

www.logicalclocks.com