MLSQL: Machine Learning Meets SQL

V. Chakraborty and G. Muktadir

CMPS 203, Spring 2019

30 May, 2019

Outline

Motivation

- Objective
  - A vast amount of data around us these days (the usual Big Data spiel!)
  - Machine Learning, a lot like the President's impeachment
    - Everyone is excited about it
    - Everyone is talking about it
    - No one (very few) is really doing it!
  - Popularity of SQL - extremely commonplace

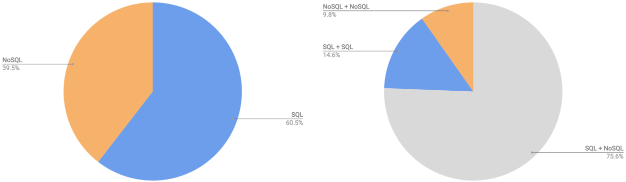

Figure: Prevalence of SQL Databses

Motivation

- Subjective
  - Python - extremely popular, host of libraries
  - SQLite
    - Portability
    - Throughput
    - Popularity
  - Popularity of SQL - extremely commonplace

Figure: SQLite usage in industry

Motivation

- What we found helpful?
  - Theoretical foundation of Relation DB model – long tested and still popular
  - Experience with
    - Tenserflow
    - Scikit Learn
    - SQL and no SQL databases
    - Pytorch
    - Theano
    - Data Analysis
    - Machine Learning
  - Distributed Systems

Overview

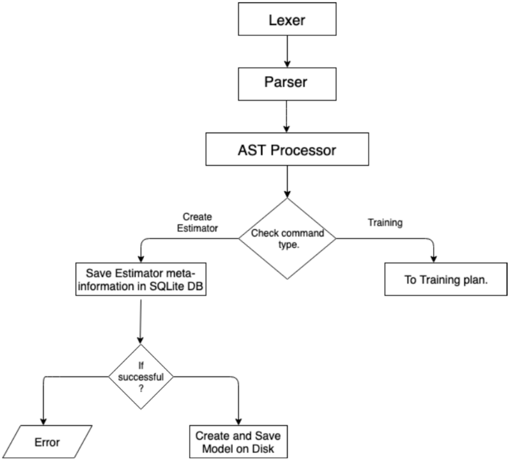

Figure 2: Execution flow in MLSQL

Figure: MLSQL

Overview
Training

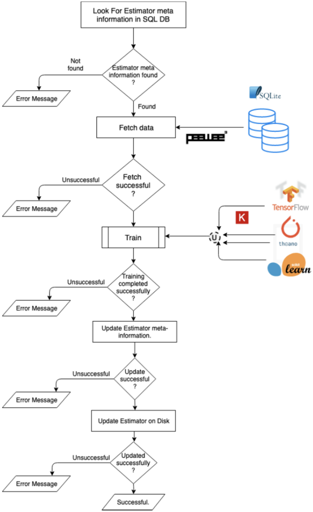

Figure: Flow to train in MLSQL

Example
Linear Regression

- **Create a model**

```
CREATE ESTIMATOR salaryPred
TYPE LR FORMULA $salary~years~...$;
```

  - attributes in FORMULA must be names of columns in tables of the dataset that will be used.

- **Create a training profile**

```
CREATE TRAINING PROFILE salaryProfile
WITH [ SELECT * FROM salary];
```

- **Select the database**

```
USE 'data/salarydb.db';
```

- **Training an estimator with a training profile**

```
TRAIN salaryPred WITH TRAINING PROFILE
salaryProfile;
```

Syntax
"Grammar Rule"

- Estimator

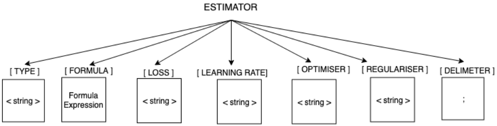

Figure: Estimator Attributes

- Training Profile

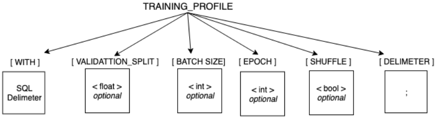

Figure: Training Profile Attributes

So what's happening?

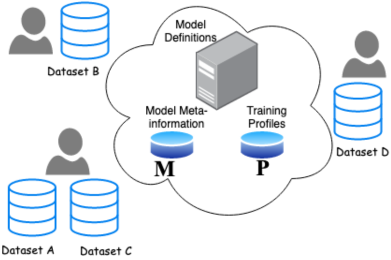

Figure: MLSQL - Overview

- Create model. Store meta-information in database M and the actual model to disk.
- Create training profile. Store profile in database P.
- Use a model from M with profile from P on a dataset of the client's choice.

A Specific Design Choice

Where to place the model/estimator?

- Two alternatives
  - Place the model inside the database
    - Better portability
  - Place it outside the database
    - Better reusability
    - Use the same model on different datasets/databases
    - Share the same model
    - Replicate model for production
- We place it outside the database
  - We foresee a setting in which MLSQL is used to create a model and used with different data-sets.
  - Trade portability for reusability
  - *Training Profile can be created by a domain expert. But the dataset can be changed by someone who is not.*

Other Design Choices

- Why SQL?
  - Recent progress – Tensor Flow API for javascript users
  - We are providing ML to SQL users

    ALL

  - Next... ML for HTML and LaTeX users !!
- SQLite3 and Pewee ORM
  - "virtual object database" that can be used from within the programming language

Is this idea completely new?

Novelty

- Surely not! However, ours is pretty dope
  - opensource
  - No "new" language
  - Abstracts using ML into two steps
    - Model Engineering (for trained experts)
    - Model Implementation (for everyone)
  - Try to minimise "shoot-in-the-dark" trend of machine learning.
- Here are some tools that inspired us...
  - Uber's Queryparser [1]
    - no ML but helped us conceive the design of our parser
  - Google's BigQuery [2] used in conjunction with MapReduce
    - serverless service
    - complicated code
    - expensive

Novelty
Continued...

- Microsoft's SQL Server 2017 [3]
  - separate core language which is quite different from other DB applications
  - expensive
  - companies that want to upgrade need to teach current employees how to work with the application
  - .NET framework dependent

Bibliography I

Queryparser Uber Engineering. https://eng.uber.com/queryparser/. Accessed: 2019-05-23.

Google AI BigQuery. https://cloud.google.com/bigquery/what-is-bigquery. Accessed: 2019-05-23.

Microsoft Sql Server ML Services. https://cloudblogs.microsoft.com/sqlserver/2017/09/26/in-database-machine-learning-in-sql-server-2017/. Accessed: 2019-05-23.

Thank you!