MLbase

Denny Britz, John Duchi, Michael J. Franklin, Rean Griffith, Joseph Gonzalez, Michael I. Jordan, Tim Kraska, Xinghao Pan, Virginia Smith, Evan Sparks, Ameet Talwalkar, Andre Wibisono

<www.mlbase.org>
Motivation
Collecting Data Is Easier Than Ever Before
My Personal Motivation: Proof
The Problem

**What you want to do**

- Build a Classifier for X

**What you have to do**

- Learn the internals of ML classification algorithms, sampling, feature selection, X-validation,....
- Potentially learn Spark/Hadoop/...
- Implement 3-4 algorithms
- Implement grid-search to find the right algorithm parameters
- Implement validation algorithms
- Experiment with different sampling-sizes, algorithms, features
- ....

and in the end

**Ask For Help**
Do We All Need To Be ML-Experts
A Declarative Approach to ML

SQL

Result
A Declarative Approach to ML

SQL > Result

MQL > Model
Use Cases

Supervised Classification: ALS Prediction

```javascript
var X = load("als_clinical", 2 to 10)
var y = load("als_clinical", 1)
var (fn-model, summary) = top(doClassify(X, y), 5min)
```

Unsupervised Feature Extraction: Twitter

```javascript
var G = loadGraph("twitter_network")
var hubs-nodes = findTopKDegreeNodes(G, k = 1000)
var text-features = textFeaturize(load("twitter_tweet_data"))
var T-hub = join(hub-nodes, "u-id", text-features, "u-id")
findTopFeatures(T-hub)
```
MLbase Architecture

- Binders full of algorithms allows to add more operators
- Statistics about algorithms and data
- Distributed Runtime build for fast (in-memory) iteration
- No single correct answer – instead approximation and continuous refinement
- Adaptive Optimizer estimates runtime and quality improvement
- Result

User

Declarative ML Task
(result: e.g., fn-model & summary)

Master Server

ML Contract + Code

ML Developer

Parser

LLP

COML (Optimizer)

Parser

Executor/Monitoring

Distributed Runtime

...
Binders Full of Algorithms
Binders Full of Algorithms

Implementation
Using high-level patterns provided by MLbase

Contract
- Type (e.g., classification)
- Parameters
- Runtime (e.g., O(n))
- Input-Specification
- Output-Specification
- ...

ML Developer
Today: Half-Full Binders

Common to state-of-the-art algorithms

• SVMs, Logistic Regression, Naïve Bayes, LogitBoost, Linear Regression, Ridge Regression, LASSO, Matrix Factorization via SGD, DFC, K-Means, DP-Means
• More to come
• Standalone implementations on Spark (soon!)
var X = load("als_clinical", 2 to 10)
var y = load("als_clinical", 1)
var (fn-model, summary) =
top(doClassify(X, y), 10min)
var $X = \text{load}(\text{"als\_clinical"}, 2 \text{ to } 10)\$

var $y = \text{load}(\text{"als\_clinical"}, 1)\$

var (fn\_model, summary) = \text{top}(\text{doClassify}(X, y), 10\text{min})\$
var X = load("als_clinical", 2 to 10)
var y = load("als_clinical", 1)
var (fn-model, summary) =
top(doClassify(X, y), 10min)
Possible Optimizations (classification)

**Relational Optimizations** (Top-K Pushdown, Join-Ordering, ...)

**Static ML Selection Rules**
- Imbalance of labels
- SVMs are more sensitive to the scale-parameter than AdaBoost to rounds
- If SVM → normalize data between [-1, 1]
- If data contains outliers → pre-clean data or forego AdaBoost
- ...

**Run-Time Optimization Rules**
- Caching: If 2nd run and deterministic, start with previously most successful model
- Set sample-size to fit Input-Data as well as intermediate result in memory
- Partition data according to cross-validation
- ...

→ **Cost-based Optimization Rules**
- Expected quality improvement based on the history
- Consider cost of pre-cleaning, normalization, algorithm complexity,...
  - ...
Why Optimize?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM</th>
<th>AdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original</td>
<td>scaled</td>
</tr>
<tr>
<td>a1a</td>
<td>82.93</td>
<td>82.93</td>
</tr>
<tr>
<td>australian</td>
<td>85.22</td>
<td>85.51</td>
</tr>
<tr>
<td>breast</td>
<td>70.13</td>
<td>97.22</td>
</tr>
<tr>
<td>diabetes</td>
<td>76.44</td>
<td>77.61</td>
</tr>
<tr>
<td>fourclass</td>
<td>100.00</td>
<td>99.77</td>
</tr>
<tr>
<td>splice</td>
<td>88.00</td>
<td>87.60</td>
</tr>
</tbody>
</table>
Why Optimize?

SVM

AdaBoost
Model Improvement over Time
Summary

• MLbase is a first declarative machine-learning system

• It simplifies ML in the same way as databases simplify data management

• MLBase is currently under heavy-development, but a first prototype is up and running:
  – 1st initial language for classification, clustering, and matrix factorization
  – Rule-based optimizer
  – Distributed algorithms for: k-means clustering, LogitBoost, various matrix factorization algorithms and support vector machines.
  – Spark and GraphLab on Spark as the run-time

• We plan to release a first prototype by this summer

  tim_kraska@brown.edu
  <www.mlbase.org>