Distributed ML in Apache® Spark™

Joseph K. Bradley

June 24, 2016

Who am I?

Apache Spark committer & PMC member

Software Engineer @ Databricks

Ph.D. in Machine Learning from Carnegie Mellon

Apache Spark

- General engine for big data computing
- Fast
- Easy to use
- APIs in Python, Scala, Java & R

Open source

- Apache Software Foundation
- 1000+ contributors
- 200+ companies & universities

Spark SQL | Streaming | MLlib | GraphX

APACHE Spark™

Largest cluster:
8000 Nodes (Tencent)

NOTABLE USERS THAT PRESENTED AT SPARK SUMMIT 2015 SAN FRANCISCO

Source: Slide 5 of Spark Community Update

NOVARTIS
UBER
AUTODESK
YAHOO!
salesforce
Microsoft
airbnb
Capital One
Alibaba.com
NBCUniversal
CISCO
Telefónica
Adobe
TOYOTA
Goldman Sachs
ebay
verizon
Palantir
NETFLIX
THOMSON REUTERS
NTT DaTa

Databricks

We're hiring!

Founded by the creators of Apache Spark

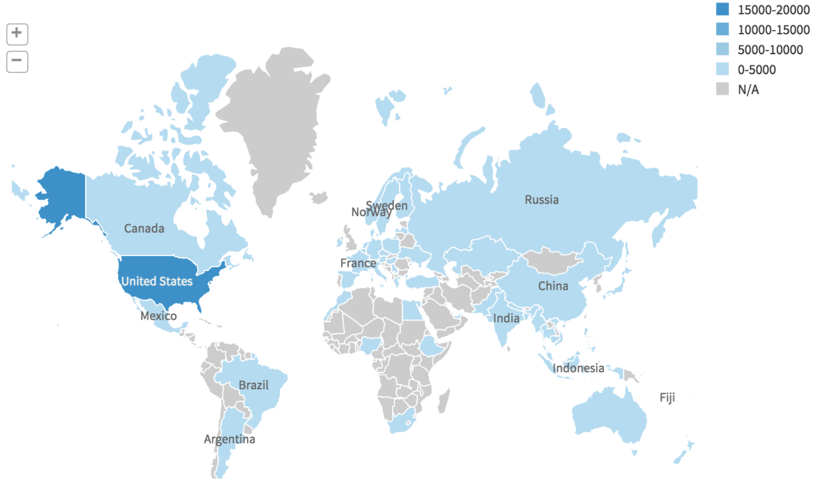
Offers hosted service

- Spark on EC2
- Notebooks
- Visualizations
- Cluster management
- Scheduled jobs

Mobile Devices by Geography (Sample Data)

This is a world map of number of mobile phones by country from a sample dataset

```
> select m.ClientID, c.CountryCode3, m.DeviceMake
  from mobile_sample m
     join countrycodes c
        on m.Country = c.Country
```



This talk: DataFrames in MLlib

Common issues within Big ML projects

- Custom, strict data format
- Library encourages developing via scripts
- Lots of work on low-level optimizations
- Hard to bridge R&D – Production gap
- Single-language APIs

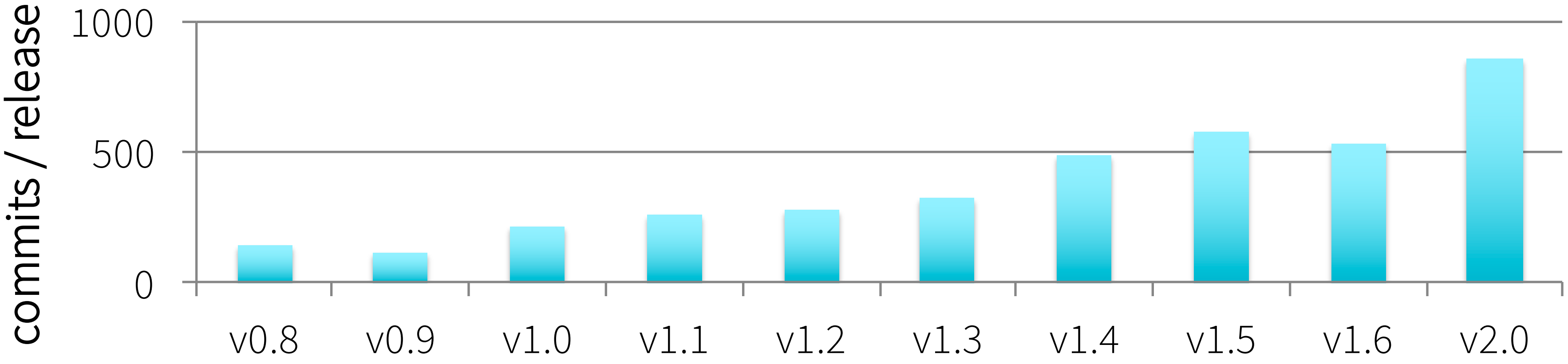
MLlib: Spark's ML library

Goals
- Scale-out ML
- Standard library
- Extensible API

Data utilities
- Featurization
- Statistics
- Linear algebra

Learning tasks
- Classification
- Regression
- Recommendation
- Clustering
- Frequent itemsets

Workflow utilities
- Model import/export
- Pipelines
- DataFrames
- Cross validation



databricks

Spark DataFrames & Datasets

dept	age	name
Bio	48	H Smith
CS	34	A Turing
Bio	43	B Jones
Chem	61	M Kennedy

Data grouped into named columns

```
data.groupBy("dept").avg("age")
```

DSL for common tasks

- Project, filter, aggregate, join, …
- 100+ functions available
- User-Defined Functions (UDFs)

Datasets: Strongly typed DataFrames

This talk: DataFrames in MLlib

Data sources & ETL

ML Pipelines

Under the hood: optimizations

Model persistence

Multiple language support

Data sources & ETL

Data scientists spend 50-80% of their time on data munging.*

DataFrames support easy manipulation of big data

- Standard DataFrame/SQL ops
- Methods for null/NaN vals
- Statistical methods
- Conversions: R data.frame, Python Pandas

Many data sources

built-in

Parquet
JDBC
{ JSON }
CSV
HIVE
MySQL
PostgreSQL
HDFS
amazon web services S3
H2

external

AVRO
dBase
APACHE HBASE
cassandra
amazon web services Amazon Redshift
elasticsearch.

and more …

* Lohraug. "For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights." NYTimes, 8/18/2014.

ML Pipelines

DataFrames: unified ML dataset API

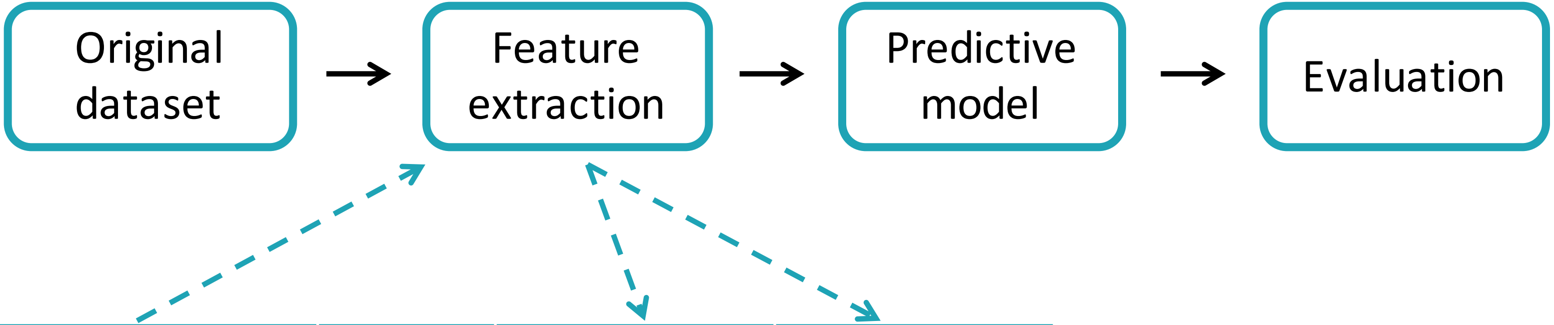
- Flexible types
- Add & remove columns during Pipeline execution

Load data

Original dataset → Feature extraction → Predictive model → Evaluation

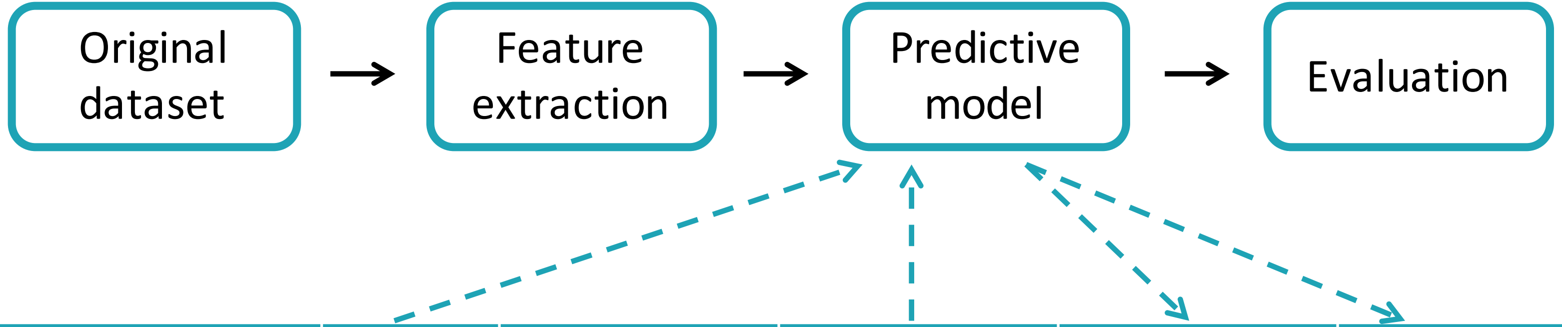
Text	Label
I bought the game...	4
Do NOT bother try...	1
this shirt is aweso...	5
never got it. Seller...	1
I ordered this to...	3

Extract features



Text	Label	Words	Features
I bought the game...	4	“i", “bought”,...	[1, 0, 3, 9, ...]
Do NOT bother try...	1	“do”, “not”,...	[0, 0, 11, 0, ...]
this shirt is aweso...	5	“this”, “shirt”	[0, 2, 3, 1, ...]
never got it. Seller...	1	“never”, “got”	[1, 2, 0, 0, ...]
I ordered this to...	3	“i”, “ordered”	[1, 0, 0, 3, ...]

Fit a model



Text	Label	Words	Features	Prediction	Probability
I bought the game...	4	“i", “bought”,...	[1, 0, 3, 9, ...]	4	0.8
Do NOT bother try...	1	“do”, “not”,...	[0, 0, 11, 0, ...]	2	0.6
this shirt is aweso...	5	“this”, “shirt”	[0, 2, 3, 1, ...]	5	0.9
never got it. Seller...	1	“never”, “got”	[1, 2, 0, 0, ...]	1	0.7
I ordered this to...	3	“i”, “ordered”	[1, 0, 0, 3, ...]	4	0.7

Evaluate

Original dataset → Feature extraction → Predictive model → Evaluation

Text	Label	Words	Features	Prediction	Probability
I bought the game...	4	"i", "bought",...	[1, 0, 3, 9, ...]	4	0.8
Do NOT bother try...	1	"do", "not",...	[0, 0, 11, 0, ...]	2	0.6
this shirt is aweso...	5	"this", "shirt"	[0, 2, 3, 1, ...]	5	0.9
never got it. Seller...	1	"never", "got"	[1, 2, 0, 0, ...]	1	0.7
I ordered this to...	3	"i", "ordered"	[1, 0, 0, 3, ...]	4	0.7

ML Pipelines

DataFrames: unified ML dataset API

- Flexible types
- Add & remove columns during Pipeline execution
- Materialize columns lazily
- Inspect intermediate results

DataFrame optimizations

Catalyst query optimizer

- Predicate pushdown
- Join selection
- …

Project Tungsten

- Memory management
  - Off-heap
  - Avoid JVM GC
  - Compressed format
- Code generation
  - Combine operations into single, efficient code blocks

Under the hood: optimizations

Current use of DataFrames

- API
- Transformations & predictions

Feature transformation & model prediction are phrased as User-Defined Functions (UDFs)
→ Catalyst query optimizer
→ Tungsten memory management + code generation

Whole-stage code generation

- Fuse across multiple operators

Implementations on DataFrames

Prototypes

- Belief propagation
- Connected components

Current challenge: DataFrame query plans do not have iteration as a top-level concept

Eventual goal: Port all ML algorithms to run on top of DataFrames → speed & scalability

ML persistence

Data Science

Prototype (Python/R)
Create model

Software Engineering

Re-implement model for production (Java)
Deploy model

ML persistence

Data Science

Prototype (Python/R)
Create Pipeline

- Extract raw features
- Transform features
- Select key features
- Fit multiple models
- Combine results to make prediction

Software Engineering

Re-implement Pipeline for production (Java)
Deploy Pipeline

- Extra implementation work
- Different code paths
- Synchronization overhead

With ML persistence...

Data Science

Prototype (Python/R)
Create Pipeline

Software Engineering

Load Pipeline (Scala/Java)
`Model.load("s3n://…")`
Deploy in production

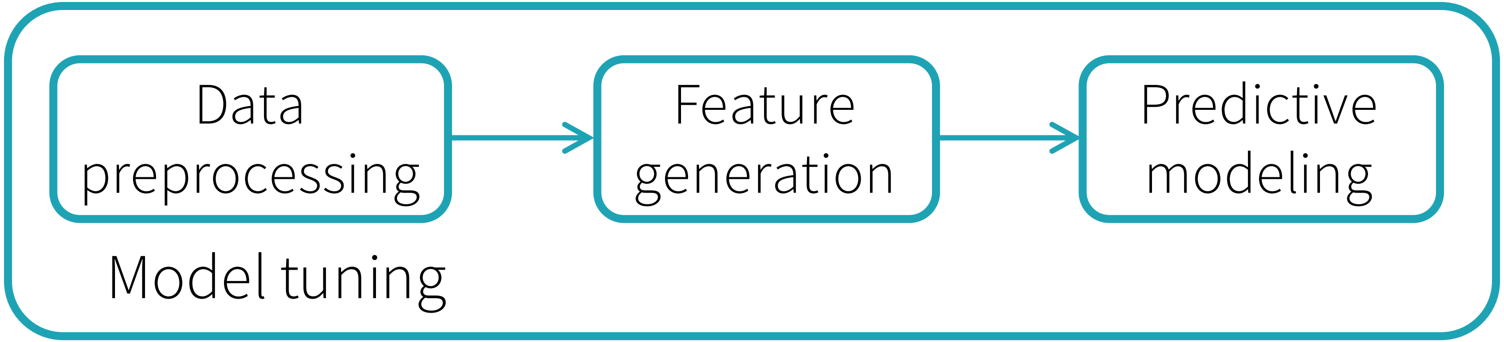
Persist model or Pipeline:
`model.save("s3n://...")`

ML persistence status

Single implementation for all Spark language APIs: Scala, Java, Python, R

	"recipe" Unfitted	"result" Fitted
Model	✓	✓
Pipeline	✓	✓

Supported in MLlib's RDD-based API



ML persistence status

Near-complete coverage in all Spark language APIs

- Scala & Java: complete
- Python: complete except for 2 algorithms
- R: complete for existing APIs

Single underlying implementation of models

Exchangeable data format

- JSON for metadata
- Parquet for model data (coefficients, etc.)

Multiple language support

APIs in Scala, Java, Python, R

- Scala (& Java): implementation
- Python & R: wrappers for Scala

DataFrames provide:

- Uniform API across languages
- Data serialization
  - Store data off-heap, accessible from JVM
  - Transfer to & from Python & R handled by DataFrames, not MLlib

Summary: DataFrames in MLlib

Data sources & ETL

ML Pipelines

Under the hood: optimizations

Model persistence

Multiple language support

Research & development topics

- Query optimization for ML/Graph algorithms
  - Caching, communication, serialization, compression
- Iteration as a first-class concept in DataFrames
- Optimized model tuning
- Spark + GPUs
- Asynchronous communication within Spark

What's next?

Prioritized items on the 2.1 roadmap JIRA (SPARK-15581):

- Critical feature completeness for the DataFrame-based API
  - Multiclass logistic regression
  - Frequent pattern mining
- Python API parity & R API expansion
- Scaling & speed for key algorithms: trees, forests, and boosting

GraphFrames

- Release for Spark 2.0
- Speed improvements (join elimination, connected components)

Get started

Get involved

- JIRA http://issues.apache.org
- mailing lists http://spark.apache.org
- Github http://github.com/apache/spark
- Spark Packages http://spark-packages.org

Many thanks to the community for contributions & support!

Learn more

- What's coming in Apache Spark 2.0 http://databricks.com/blog/2016/06/01
- MOOCs on EdX http://databricks.com/spark/training

Try out Apache Spark 2.0 preview in Databricks Community Edition

http://databricks.com/ce

Thank you!

Twitter: @jkbatcmu

We're hiring!
http://databricks.com/careers

databricks™