Apache Mahout
An Extendable Machine Learning Framework for Flink and Spark (and others)
About Me

Trevor Grant

PMC Apache Mahout Project
PPMC Apache Streams-Incubator

Open Source Evangelist, IBM

@rawkintrevo

rawkintrevo@apache.org

http://rawkintrevo.org

Formerly “Data Scientist” in Media Strategy and CPG
This talk is about Community, not Code
Community over code

NO:

● Features etc. of new release
● How to use Mahout
● Examples of cool things to do in Mahout

YES:

● The story of a project that has faced some unique challenges
● How we have repositioned our project to be more contributor friendly
● How we are actively evangelizing and seeking new contributors
Brief History

Began as subproject of Lucene-2008

Graduated to TLP in May 2010

Feb 2014 Mahout 0.9 released (last Map-Reduce Version)

Several Major Hadoop Vendors “froze” here

Lots of blogs/tutorial with high search relevance still point at Map-Reduce based Mahout implementations
Rebranding
Mahout Evangelism: Rebranding

SAY "MAHOUT IS BUILT FOR RUNNING ON MAP-REDUCE"

ONE MORE TIME, I DARE YOU
“Marketing Funnel”
Evangelism: Rebranding

**Problem:** High brand-recognition… for something we don’t do anymore

**Opportunity:** Now relevant to many more areas- GPUs, Spark, Flink, etc.

**Solution:** Talks and Blog Posts

**Corollary:** New awareness, and changing perceptions
Mahout Evangelism: Talks (Fall ‘16)

Suneel Marthi and Trevor Grant did a Mahout on Flink talk at Flink Forward 2016, Berlin, Germany - September 13, 2016

Suneel Marthi did a Mahout talk at Department of Theoretical Physics, Fritz-Haber Institut der Max Planck Gessellschaft, Berlin, Germany - September 16, 2016

Suneel Marthi did a ‘Distributed Machine Learning with Apache Mahout’ talk at Big Data Ignite, Grand Rapids, Michigan - September 30, 2016

Sebastian Schelter presented a poster at Machine Learning Systems Workshop, NIPS 2016 Dec 10, 2016 - “Samsara: Declarative
Mahout Evangelism: Talks (Fall ‘16) continued


Andrew Palumbo presented “Apache Mahout: Beyond MapReduce” at the Orange County Big Data Meetup, October, 2016.

Trevor Grant presented: “Apache Mahout?! What’s Next!” At
   a. Chicago Hadoop Users Group, October 2016
   b. Seattle Data Science Meetup, December 2016
   c. San Diego Big Data Meetup, December 2016
   d. Austin Data Meetup, December 2016
   e. DFW Data Science Meetup, December 2016

(^^ Thank you IBM)

Andrew Musselman presented: “Apache Mahout?! What’s Next!” at Seattle Data Science Meetup, December 2016

Suneel Marthi presented: “Native and Distributed Machine Learning with Apache Mahout” - Apache Big Data Europe 2016, Nov 13 2016, Seville, Spain
Mahout Evangelism: Talks (Spring ‘17)

APACHE MAHOUT'S NEW RECOMMENDER ALGORITHM AND USING GPUS TO SPEED MODEL CREATION Pat Ferrel, Andy Palumbo. GPU Technology Conference. Silicon Valley, CA- May 11, 2017

EXTENDING MAHOUT-SAMSARA LINEAR ALGEBRA DSL TO SUPPORT GPU CLUSTERS Suneel Marthi, Trevor Grant. GPU Technology Conference. Silicon Valley, CA- May 11, 2017


An Apache Based Intelligent IoT Stack for Transportation Trevor Grant, Joe Olsen. ApacheCon IoT. Miami, FL- May 18, 2017

(+2 at ApacheCon/Apache Big Data but last minute speaker had conflict)


Introduction to Online Machine Learning Algorithms Trevor Grant. Dataworks Summit. San Jose, CA- June 15, 2017
Mahout Evangelism: Blog Posts

Post-Map Reduce

Introducing Precanned Algorithms in Apache Mahout - May 2, 2017. Trevor Grant
https://rawkintrevo.org/2017/05/02/introducing-pre-canned-algorithms-apache-mahout/

Getting Started with Apache Mahout - April 25, 2017. Trevor Grant

Correlated Cross-Occurrence (CCO): How to make data behave as we want - December 1, 2016. Pat Ferrel
http://actionml.com/blog/cco
Mahout Evangelism: Blog Posts

Map Reduce

**Mahout ItemSimilarity** - Feb 1, 2017.
http://kathytech.info/?tag=apache-mahout

**The Year in Machine Learning (Part Two)** - Jan 2, 2017.
https://thomaswdinsmore.com/tag/apache-mahout/
Evangelism: Social Engagement
Evangalism
Relevant Features
Extremely relevant features.

**Problem:** GPU Servers can only support 8 graphics cards

**Opportunity:** New Mahout “Native Solvers” allows users to leverage a distributed engine (like Apache Spark) and scale to an arbitrary number of graphics cards (one or two per node on commodity hardware)

**Solution:** Make this a central part of our offering, market the heck out of it.

**Corollary:** Mahout’s “competitor” isn’t MLLib (which we blow away), its TensorFlow and MXNet, as well as vendors of GPU servers (frenemies) and we have a much better product, whose sole purpose isn’t to onramp customers onto a particular platform.
Battle of the Distributed Machine Learning

MXNET AND TENSORFLOW ARE FIGHTING TILL THE DEATH

I'LL JUST WAIT HERE FOR THE WINNER
Other Killer New Features

Will touch on them elsewhere-

Algo Framework / Precanned Algos

Zeppelin Integration

Mahout-Samsara (Mathematically Expressive Scala DSL)

Etc.
Evangalism

Awareness

Consideration

Download

User

Contributor
New Image
Website

What is Apache Mahout?

The Apache Mahout™ project’s goal is to build an environment for quickly creating scalable performant machine learning applications.

Apache Mahout software provides three major features:
- A simple and extensible programming environment and framework for building scalable algorithms
- A wide variety of premade algorithms for Scala + Apache Spark, H2O, Apache Flink
- Samsara, a vector math experimentation environment with R-like syntax which works at scale

Read an Overview of programming a Mahout Samsara Application, learn How To Contribute to Mahout, report an issue, bug, or suggestion in our JIRA, see the Samsara bindings for Scala and Spark, and contact us on our mailing lists.

13 May 2017 - Apache Mahout website beta release
Docs available here

17 April 2017 - Apache Mahout 0.13.0 released
Mahout 0.13.0 includes
- In-core matrices backed by ViennaCL 3 providing in some cases speedups of an order of magnitude.
- A JavaCPP bridge to nativeGPU operations in ViennaCL
- A simple and extensible programming environment and framework for building scalable algorithms
- A wide variety of premade algorithms for Scala + Apache Spark, H2O, Apache Flink
- Samsara, a vector math experimentation environment with R-like syntax which works at scale

Latest release version 0.12.2 has
Apache Mahout Samsara Environment includes
- Distributed Algebraic optimizer
- R-Like DSL, Scala API
- Linear algebra operations
- Ops are extensions to Scala
- IScale REPL, based interactive shell
- Integrates with compatible libraries like MLlib
- Runs on distributed Spark, H2O, and Flink
- fastutil to speed up sparse matrix and vector computations
- Matrix to tensor conversions for integration with Apache Zeppelin

Apache Mahout Samsara Algorithms include
- Stochastic Singular Value

Download Mahout

Twitter
Nah, the old website is good.
New Image

Problem: Old Website

Opportunity: Websites are easy (technically- design on the other hand...)

Solution: Reboot website with Jekyll Bootstrap (and less emphasis on Map-Reduce)

Corollary: Much easier for committers and contributors to update website, add tutorials, etc. Encourages (requires) new features to be submitted with good docs.
Ordinary Least Squares Regression

About

The OrdinaryLeastSquares regressor in Mahout implements a closed-form solution to Ordinary Least Squares. This is in stark contrast to many “big data machine learning” frameworks which implement a stochastic approach. From the users perspective this difference can be reduced to:

- **Stochastic** - A series of guesses at a line of best fit.
- **Closed Form** - A mathematical approach has been explored, the properties of the parameters are well understood, and problems which arise (and the remedial measures, exist. This is usually the preferred choice of mathematicians/statisticians, but computational limitations have forced us to resort to SGD.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>calcCommonStatistics</td>
<td>Calculate commons statistics such as Coefficient of Determination and Mean Square Error</td>
<td>true</td>
</tr>
<tr>
<td>calcStandardErrors</td>
<td>Calculate the standard errors (and subsequent “t-scores” and “p-values”) of the beta estimates</td>
<td>true</td>
</tr>
<tr>
<td>addIntercept</td>
<td>Add an intercept to X</td>
<td>true</td>
</tr>
</tbody>
</table>

Example

In this example we disable the “calculate common statistics” parameters, so our summary will NOT contain the coefficient of determination (R-squared) or Mean Square Error.

```java
import org.apache.mahout.math.algorithms.regression.OrdinaryLeastSquares

val data = drmParallelize(dense[
  (2, 2, 18.5, 18, 29.500601), // Apple Cinnamon Cheerios
  (1, 1, 12, 12, 18.042851), // Cap'n'Crunch
  (1, 1, 12, 13, 22.736446), // Cocoa Puffs
]...
New Image

**Problem:** Logo no longer relevant + math nerds aren’t great artists

**Opportunity:** Everyone has friends!

**Solution:** Actively searching for proposal idea for new logo- discussed on mailing lists

**Corollary:** Part of rebranding effort (hard to dissociate from Mapreduce when your logo has the yellow elephant)
New Logo (?)

This is what happens when math nerds design logos at 3 am.

I still like it better than our current one.

(These memes are what happens when math nerds design memes at 1am..)
Apache Zeppelin integration
Huge win

Scala has weak visualization.

R and Python have great visualization!

Zeppelin allows user to hand off variables between interpreters in notebook.

Do work in Mahout (Scala) - Plot in R/Python!

Mahout Gaussian DRM plotted in R
Huge win- kinda wonky install.

A script exists for assisting with install… needs update for 0.13.0
https://issues.apache.org/jira/browse/ZEPELLIN-2417

Major Issue- Zeppelin out of the box supports Scala 2.11/Spark 2.1

Mahout 0.13.0 will build, but no binaries exist.

User must either build Mahout (for Spark 2.1) or Build Zeppelin (for Spark 1.6)

Fix coming soon in 0.13.1 (profiles and binaries for Spark 2.1/Scala 2.11)
http://mahout.apache.org/docs/0.13.1-SNAPSHOT/tutorials/misc/mahout-in-zeppelin/
Value

It’s fun to integrate with other ASF products.

Ton’s of users- esp data science users.

Good exposure to Scala-Spark based Mahout for new users. (Counts as a download!)
Evangelism
Mathematically Expressive Scala DSL
Big Math

Problem: May not know Scala/Apache Spark/Apache Flink

Opportunity: Do know R

Solution: Create an abstracted language- mathematically expressive Scala DSL
implicit val sdc: org.apache.mahout.sparkbindings.SparkDistributedContext = sc2sdc(sc)

val A = drmWrap(rddA)
val B = drmWrap(rddB)

val C = A.t %*% A + A %*% B.t

C is a RDD[Int, org.apache.mahout.Vector]

Also have truly distributed matrix decompositions.
Why is anyone using Tensorflow/MxNet?

IF SCALA WAS MORE EXPRESSIVE THAN PYTHON

WHY WOULD ANYONE USE PYTHON?

Hint: Google and Amazon spend a lot of money luring you into their platform
Big Math

Venn Diagram-
Interest, Math Qualified, Can write for distributed frameworks
Moved Math Qualified circle over- now overlaps more with distributed frameworks
Engine Neutrality
Engine Neutrality

**Problem:** Distributed engines come and go (we learned this first)

**Opportunity:** We learned this first!

**Solution:** Create Engine Neutral Libraries which can bind to new engines

**Corollary:** Implement algorithm once- run it anywhere*
We don’t actually have Beam bindings (yet), this is just for lulz

HEARD YOU LIKE ABSTRACTING DISTRIBUTED ENGINES

SO I WROTE BEAM BINDINGS FOR MAHOUT NOW YOUR ABSTRACTION IS ABSTRACTED
Engine Neutrality

Awareness

Consideration

Download

User

Contributor
Algorithm Framework
Attracting Contributors

**Problem:** Very small pool of qualified contributors

**Opportunity:** Mathematically expressive scala makes it easy to write and review “math part”

**Solution:** Create templates and tutorials showing users how to add algorithm
Drop your algorithms in easily

Algorithms are easy to compose in Mahout (as easy* as R, often can use R implementation for guidance)

Scala classes / package layout may still be overwhelming for our target users
Algorithm Template

```java
class Foo[K] extends RegressorFitter[K] {

    def fit(drmX : DrmLike[K],
             drmTarget: DrmLike[K],
             hyperparameters: (Symbol, Any)*): FooModel[K] = {
        /**<
         * Normally one would have a lot more code here.
         */
        var model = new FooModel[K]
        model.summary = "This model has been fit, I would tell you more interesting things- if this
        model
        }
    }

    class FooModel[K] extends RegressorModel[K] {

        def predict(drmPredictors: DrmLike[K]): DrmLike[K] = {
            drmPredictors.mapBlock(1) {
                case (keys, block: Matrix) => {
                    var outputBlock = new DenseMatrix(block.nrow, 1)
                    keys -> (outputBlock += 1.0)
                }
            }
        }
    }
}
```
Encourage using dev

A lot of hand holding with first time contributors- encourage them to pay it forward.
New Algo Contributors in Progress

DBSCAN (GSoC project)

Regression F-Test

K-Means Classification
Engine Neutrality

- Awareness
- Consideration
- Download
- User
- Contributor
How this actually progressed...
First The Math and Engine Neutrality

2014-2016
Then we started our evangelism

- Awareness
  - Consideration
  - Download
  - User
    - Contributor

2014-ongoing, big spike recently
Then the GPU work and algorithm framework
Still finishing up the website...

Q2 2017- ongoing
Hooray!

- Awareness
- Consideration
- Download
- User
- Contributor
What’s Next
Building Algorithms Framework

Framework in place which encourages users to contribute algorithms (already paying off)

Seeking to grow “pre-canned” algorithms collection between now and v 0.14.0

Eventually a “CRAN” like repository of algorithms for Mahout.
More work on GPUs

We consider our GPU support a HUGE differentiator among ML packages native to distributed engines (MLLib, FlinkML, etc).

Still opportunities for optimization-

Recent benchmark on unoptimized (still technically PR) CUDA bindings show “significant” speed up on sparse Matrix multiplication
30%

Speed up On Sparse-Sparse Matrix Multiplication on AWS GPU enabled Spark Cluster

Kind of a big deal.
More Engine Bindings

Create template engine bindings- even if not optimized.

Tutorials on writing new engine bindings.

We feel this is also a huge differentiator.

Possibly “Community” supported engine bindings, not officially supported- but in the trunk, attract “drive-by” contributions from other projects.
Conclusion
Awakening the Giant

Mahout has quietly undergone huge transformation from Map Reduce / Java based Machine Learning to Mathematically Expressive Scala / Engine Neutral / GPU Accelerated

Need to let everyone know- I mean you- go tell your friends and tweet and write a blog.
“I want you so badly” - The Beatles

I WANT YOU
TO TELL ALL YOUR INTERNET “FRIENDS”
ABOUT MAHOUT

I WANT YOU
TO CONTRIBUTE

I WANT YOU
TO JOIN THE MAILING
LISTS
Questions?