"I want to die on Mars but not on impact"
- Elon Musk, interview with Chris Anderson

"There are no facts, only interpretations." - Friedrich Nietzsche

"The shrewd guess, the fertile hypothesis, the courageous leap to a tentative conclusion – these are the most valuable coin of the thinker at work" -- Jerome Seymour Bruner

"If you torture the data long enough, it will confess to anything." – Hal Varian, Computer Mediated Transactions

------ We are not going to hang data by its legs!
ADVANCED: DATA SCIENCE WITH APACHE SPARK

Data Science applications with Apache Spark combine the scalability of Spark and the distributed machine learning algorithms.

This material expands on the “Intro to Apache Spark” workshop. Lessons focus on industry use cases for machine learning at scale, coding examples based on public data sets, and leveraging cloud-based notebooks within a team context. Includes limited free accounts on Databricks Cloud.

Topics covered include:

- Data transformation techniques based on both Spark SQL and functional programming in Scala and Python.

- Predictive analytics based on MLlib, clustering with KMeans, building classifiers with a variety of algorithms and text analytics – all with emphasis on an iterative cycle of feature engineering, modeling, evaluation.

- Visualization techniques (matplotlib, ggplot2, D3, etc.) to surface insights.

- Understand how the primitives like Matrix Factorization are implemented in a distributed parallel framework from the designers of MLlib

- Several hands-on exercises using datasets such as Movielens, Titanic, State Of the Union speeches, and RecSys Challenge 2015.

Prerequisites:
- Intro to Apache Spark workshop or equivalent (e.g., Spark Developer Certificate)
- Experience coding in Scala, Python, SQL
- Have some familiarity with Data Science topics (e.g., business use cases)
Agenda

- Detailed agenda in Google doc
- https://docs.google.com/document/d/1T9AkXUmL6gDYTpAEEgsyqhfJtqGlavjnGzMqzUwDFE/edit
Goals

• **Patterns**: Data wrangling (Transform, Model & Reason) with Spark
  - Use RDDs, Transformations and Actions in the context of a Data Science Problem, an Algorithms & a Dataset

• Spend time working through MLlib

• Balance between internals & hands-on
  - Internals from Reza, the MLlib lead

• ~65% of time on Databricks Cloud & Notebooks
  - Take the time to get familiar with the Interface & the Data Science Cloud
  - *Make mistakes, experiment,*...

• Good Time for this course, this version
  - Will miss many of the gory details as the framework evolves

• Summarized materials for a 3 day course
  - Even if we don’t finish the exercises today, that is fine
  - Complete the work at home - *There are also homework notebooks*
  - Ask us questions @ksankar, @pacoid, @reza_zadeh, @mhfalaki, @andykonwinski, @xmeng, @michaelarmbrust, @tathadas
<table>
<thead>
<tr>
<th>morning</th>
<th>afternoon</th>
</tr>
</thead>
<tbody>
<tr>
<td>o  Welcome + Getting Started <em>(Krishna)</em></td>
<td>o  Ex 3 : Clustering - In which we explore Segmenting Frequent InterGalacticHoppers <em>(Krishna)</em></td>
</tr>
<tr>
<td>o  Databricks Cloud mechanics <em>(Andy)</em></td>
<td>o  Ex 4 : Recommendation <em>(Krishna)</em></td>
</tr>
<tr>
<td>o  Ex 0: Pre-Flight Check <em>(Krishna)</em></td>
<td>o  Theory : Matrix Factorization, SVD,... <em>(Reza)</em></td>
</tr>
<tr>
<td>o  DataScience DevOps - Introduction to Spark <em>(Krishna)</em></td>
<td>o  On-line k-means, spark streaming <em>(Reza)</em></td>
</tr>
<tr>
<td>o  Ex 1: MLib : Statistics, Linear Regression <em>(Krishna)</em></td>
<td></td>
</tr>
<tr>
<td>o  MLib Deep Dive – Lecture <em>(Reza)</em></td>
<td></td>
</tr>
<tr>
<td>o  Design Philosophy, APIs</td>
<td></td>
</tr>
<tr>
<td>o  Ex 2: In which we explore Disasters, Trees, Classification &amp; the Kaggle Competition <em>(Krishna)</em></td>
<td>o  Ex 5 : Mood of the Union-Text Analytics*(Krishna)*</td>
</tr>
<tr>
<td>o  Random Forest, Bagging, Data De-correlation</td>
<td>o  In which we analyze the Mood of the nation from inferences on SOTU by the POTUS <em>(State of the Union Addresses by The President Of the US)</em></td>
</tr>
<tr>
<td>o  Deepdive - Leverage parallelism of RDDs, sparse vectors, etc <em>(Reza)</em></td>
<td>o  Ex 99 : RecSys 2015 Challenge <em>(Krishna)</em></td>
</tr>
<tr>
<td></td>
<td>o  Ask Us Anything - Panel</td>
</tr>
</tbody>
</table>
Introducing:

Reza Zadeh
@Reza_Zadeh

Hossein Falaki
@mhfalaki

Michael Armbrust
@michaelarmbrust

Xiangrui Meng
@xmeng

Tathagata Das
@tathadas

Andy Konwinski
@andykonwinski

Paco Nathan
@pacoid

Krishna Sankar
@ksankar
About Me

- Chief Data Scientist at BlackArrow.tv
- Have been speaking at OSCON, PyCon, Pydata, Strata et al.
- Reviewer “Machine Learning with Spark”
- Picked up co-authorship Second Edition of “Fast Data Processing with Spark”
- Have done lots of things:
  - Big Data (Retail, Bioinformatics, Financial, AdTech...)
  - Written Books (Web 2.0, Wireless, Java, ...) 
  - Standards (Web Service, Cloud), Some work in AI
  - Guest Lecturer at Naval PG School, ...
  - Planning Masters Computational Finance or Statistics 
  - Volunteer as Robotics Judge at First Lego league World Competitions
- @ksankar, doubleclix.wordpress.com ksankar42@gmail.com
Pre-requisites

① Register & Download data from Kaggle.
   We cannot distribute Kaggle data.
   Moreover you need an account to submit entries
   a) Setup an account in Kaggle (www.kaggle.com)
   b) We will be using the data from the competition “Titanic: Machine Learning from Disaster”
   c) Download data from
      http://www.kaggle.com/c/titanic-gettingStarted

② Register for RecSys 2015 Competition
   a) http://2015.recsyschallenge.com/
Welcome +
Getting Started
Getting Started: Step 1

Everyone will receive a username/password for one of the Databricks Cloud shards. Use your laptop and browser to login there.

We find that cloud-based notebooks are a simple way to get started using Apache Spark – as the motto “Making Big Data Simple” states.

Please create and run a variety of notebooks on your account throughout the tutorial. These accounts will remain open long enough for you to export your work.

See the product page or FAQ for more details, or contact Databricks to register for a trial account.
Getting Started: Step 1 – Credentials

url: https://class01.cloud.databricks.com/
user: student-777
pass: 93ac11xq23z5150
cluster: student-777

Stuart Layton @cambridge_stu · 2h
Access to @databricks cloud for attending #SparkSummitEast! Best #swag ever?
Getting Started: Step 2

Open in a browser window, then click on the navigation menu in the top/left corner:
Getting Started: Step 3

The next columns to the right show folders, and scroll down to click on databricks_guide
Getting Started: Step 4

Scroll to open the 01 Quick Start notebook, then follow the discussion about using key features:
Getting Started: Step 5

See /databricks-guide/01 Quick Start

Key Features:

• Workspace / Folder / Notebook
• Code Cells, run/edit/move/comment
• Markdown
• Results
• Import/Export
Getting Started: Step 6

Click on the **Workspace** menu and create your own folder (pick a name):
**Getting Started: Step 7**

Navigate to / _DataScience_
Hover on its drop-down menu, on the right side:
Getting Started: Step 8

Navigate to /DataScience
Hover on its drop-down menu, on the right side:
Click Clone:
Getting Started: Step 9

Then create a *clone* of this folder in the folder that you just created:
Getting Started: Coding Exercise

Now let’s get started with the coding exercise!
We’ll define an initial Spark app in three lines of code:
Click on _00.pre-flight-check
Getting Started: Step 10

Attach your *cluster* – same as your *username*:
Introduction To Spark
**Data Science:**

The art of building a model with known knowns, which when let loose, works with unknown unknowns!

Donald Rumsfeld is an armchair Data Scientist!

<table>
<thead>
<tr>
<th>The World</th>
<th>You</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knowns</strong></td>
<td><strong>UnKnown</strong></td>
</tr>
<tr>
<td></td>
<td>Others know, you don’t</td>
</tr>
<tr>
<td></td>
<td>Facts, outcomes or scenarios we have not encountered, nor considered</td>
</tr>
<tr>
<td></td>
<td>“Black swans”, outliers, long tails of probability distributions</td>
</tr>
</tbody>
</table>

- **Known Knowns**
  - There are things we know that we know
- **Known Unknowns**
  - That is to say, there are things that we now know we don’t know
- **But there are also Unknown Unknowns**
  - There are things we do not know we don’t know

http://smartorg.com/2013/07/valuepoint19/
The curious case of the Data Scientist

- Data Scientist is multi-faceted & Contextual
- Data Scientist should be building Data Products
- Data Scientist should tell a story

Data Scientist (noun): Person who is better at statistics than any software engineer & better at software engineering than any statistician
- Josh Wills (Cloudera)

Large is hard; Infinite is much easier!
- Titus Brown

Data Scientist (noun): Person who is worse at statistics than any statistician & worse at software engineering than any software engineer
- Will Cukierski (Kaggle)
Data Science - Context

Collect
- Volume
- Velocity
- Streaming Data
- Metadata
- Monitor counters & Metrics
- Structured vs. Multi-structured
- Canonical form
- Data catalog
- Data Fabric across the organization
- Access to multiple sources of data
- Think Hybrid – Big Data Apps, Appliances & Infrastructure

Store
- Metadata
- Monitor counters & Metrics
- Structured vs. Multi-structured

Transform
- Canonical form
- Data catalog
- Data Fabric across the organization
- Access to multiple sources of data
- Think Hybrid – Big Data Apps, Appliances & Infrastructure

Model
- Flexible & Selectable
  - Data Subsets
  - Attribute sets
- Refine model with
  - Extended Data subsets
  - Engineered Attribute sets
  - Validation run across a larger data set

Deploy
- Scalable Model Deployment
- Big Data automation & purpose built appliances (soft/hard)
- Manage SLAs & response times

Data Management

Data Science
- Bytes to Business a.k.a. Build the full stack
- Find Relevant Data For Business
- Connect the Dots

Visualize
- Performance
- Scalability
- Refresh Latency
- In-memory Analytics
- Advanced Visualization
- Interactive Dashboards
- Map Overlay
- Infographics

Recommend

Predict

Explore
- Dynamic Data Sets
- 2 way key-value tagging of datasets
- Extended attribute sets
- Advanced Analytics
Data Science - Context

- Three Amigos
  - Interface = Cognition
  - Intelligence = Compute(CPU) & Computational(GPU)
  - Infer Significance & Causality

"Data of unusual size" that can't be brute forced
Day in the life of a (super) Model

Model Selection

Reason & Learn

Visualize, Recommend, Explore

Intelligence

Inference

Data Representation

Interface

Models

Attributes

Parameters

Data (Scoring)

Algorithms

Dimensionality Reduction

Feature Selection

Model Assessment

Reason & Learn

Explore

Parameter

Data (Scoring)

Features

Model Assessment
A Shift In Perspective

Analytic in the Lab
- Question-driven
- Interactive
- Ad-hoc, post-hoc
- Fixed data
- Focus on speed and flexibility
- Output is embedded into a report or in-database scoring engine

Analytic in the Factory
- Metric-driven
- Automated
- Systematic
- Fluid data
- Focus on transparency and reliability
- Output is a production system that makes customer-facing decisions

Lab = Investigative

Factory = Operational

Spark - The Stack

- Spark SQL
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

Apache Spark
RDD – The workhorse of Spark

- Resilient Distributed Datasets
  - Collection that can be operated in parallel
- Transformations – create RDDs
  - Map, Filter, …
- Actions – Get values
  - Collect, Take, …
- We will apply these operations during this tutorial
MLlib Hands-on Stats, Linear Regression
Algorithm spectrum

- **Regression**
  - Logit
  - CART
  - Ensemble: Random Forest

- **Clustering**
  - KNN
  - Genetic Algorithm
  - Simulated Annealing

- **Collab Filtering**
  - SVM
  - Kernels
  - SVD

- **NNet**
  - Boltzmann Machine
  - Feature Learning

Machine Learning → Cute Math → Artificial Intelligence
Session – 1 : MLlib - Statistics & Linear Regression

1. Notebook : 01_StatsLR-1
   ① Read car data
   ② Stats (Guided)
   ③ Correlation (Guided)
   ④ Coding Exercise-21-Template (Correlation)

2. Notebook : 02_StatsLR-2
   ① CE-21-Solution

3. Linear Regression
   ① LR (Guided)
   ② CE-22-Template((LR on Car Data))

4. Notebook : 03_StatsLR-3
   ① CE-22-Solution
   ② Explain
**Linear Regression - API**

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LabeledPoint</td>
<td>The features and labels of a data point</td>
</tr>
<tr>
<td>LinearModel</td>
<td>weights, intercept</td>
</tr>
<tr>
<td>LinearRegressionModelBase</td>
<td>predict()</td>
</tr>
<tr>
<td>LinearRegressionModel</td>
<td></td>
</tr>
<tr>
<td>LinearRegressionWithSGD</td>
<td>train(cls, data, iterations=100, step=1.0, miniBatchFraction=1.0, initialWeights=None, regParam=1.0, regType=None, intercept=False)</td>
</tr>
<tr>
<td>LassoModel</td>
<td>Least-squares fit with an L1 penalty term.</td>
</tr>
<tr>
<td>LassoWithSGD</td>
<td>train(cls, data, iterations=100, step=1.0, regParam=1.0, miniBatchFraction=1.0, initialWeights=None)</td>
</tr>
<tr>
<td>RidgeRegressionModel</td>
<td>Least-squares fit with an L2 penalty term.</td>
</tr>
<tr>
<td>RidgeRegressionWithSGD</td>
<td>train(cls, data, iterations=100, step=1.0, regParam=1.0, miniBatchFraction=1.0, initialWeights=None)</td>
</tr>
</tbody>
</table>
MLlib - Deep Dive

- Design Philosophy & APIs
- Algorithms - Regression, SGD et al
- Interfaces
Distributed Machine Learning on Spark

Reza Zadeh

@Reza_Zadeh | http://reza-zadeh.com
Outline

Data flow vs. traditional network programming
Spark computing engine
Optimization Examples
Matrix Computations
MLlib + \{Streaming, GraphX, SQL\}
Future of MLlib
Data Flow Models

Restrict the programming interface so that the system can do more automatically

Express jobs as graphs of high-level operators
  » System picks how to split each operator into tasks and where to run each task
  » Run parts twice fault recovery

Biggest example: MapReduce
Spark Computing Engine

Extends a programming language with a distributed collection data-structure
  » “Resilient distributed datasets” (RDD)

Open source at Apache
  » Most active community in big data, with 50+ companies contributing

Clean APIs in Java, Scala, Python

Community: SparkR, soon to be merged
Key Idea

Resilient Distributed Datasets (RDDs)
» Collections of objects across a cluster with user-controlled partitioning & storage (memory, disk, ...)
» Built via parallel transformations (map, filter, ...)
» The world only lets you make make RDDs such that they can be:
  
  Automatically rebuilt on failure
MLlib History

MLlib is a Spark subproject providing machine learning primitives

Initial contribution from AMPLab, UC Berkeley

Shipped with Spark since Sept 2013
**MLlib: Available algorithms**

**classification:** logistic regression, linear SVM, naïve Bayes, least squares, classification tree

**regression:** generalized linear models (GLMs), regression tree

**collaborative filtering:** alternating least squares (ALS), non-negative matrix factorization (NMF)

**clustering:** k-means||

**decomposition:** SVD, PCA

**optimization:** stochastic gradient descent, L-BFGS
Optimization

At least two large classes of optimization problems humans can solve:

» Convex

» Spectral
Optimization Example: Convex Optimization
Logistic Regression

\[
w \leftarrow w - \alpha \cdot \sum_{i=1}^{n} g(w; x_i, y_i)
\]

```scala
val points = spark.textFile(...).map(parsePoint).cache()
val w = Vector.zeros(d)
for (i <- 1 to numIterations) {
  val gradient = points.map { p =>
    (1 / (1 + exp(-p.y * w.dot(p.x)) - 1)) * p.y * p.x
  ).reduce(_ + _)
  w -= alpha * gradient
}```
Separable Updates

Can be generalized for

» Unconstrained optimization

» Smooth or non-smooth

» LBFGS, Conjugate Gradient, Accelerated Gradient methods, …
100 GB of data on 50 m1.xlarge EC2 machines

110 s / iteration

first iteration 80 s
further iterations 1 s
Behavior with Less RAM

![Bar chart showing iteration time (s) vs. % of working set in memory.

- 0%: 68.8 seconds
- 25%: 58.1 seconds
- 50%: 40.7 seconds
- 75%: 29.7 seconds
- 100%: 11.5 seconds]
Optimization Example: Spectral Program
Spark PageRank

Given directed graph, compute node importance. Two RDDs:

» Neighbors (a sparse graph/matrix)
» Current guess (a vector)

Using cache(), keep neighbor list in RAM
Spark PageRank

Using cache(), keep neighbor lists in RAM
Using partitioning, avoid repeated hashing
PageRank Results

Time per iteration (s)

- Hadoop: 171
- Basic Spark: 72
- Spark + Controlled Partitioning: 23
Spark PageRank

Generalizes to Matrix Multiplication, opening many algorithms from Numerical Linear Algebra
Distributing Matrix Computations
Distributing Matrices

How to distribute a matrix across machines?
» By Entries (CoordinateMatrix)
» By Rows (RowMatrix)
» By Blocks (BlockMatrix)  As of version 1.3

All of Linear Algebra to be rebuilt using these partitioning schemes
Distributing Matrices

Even the simplest operations require thinking about communication e.g. multiplication

How many different matrix multiplies needed?

» At least one per pair of \{Coordinate, Row, Block, LocalDense, LocalSparse\} = 10

» More because multiplies not commutative
Coffee-Break
Back at 10:45
MLlib - Hands On #2 – Kaggle Competition Predicting Titanic Survivors:

- Feature Engineering
- Classification Algorithms (Random forest)
- Submission & Leaderboard scores
"If you torture the data long enough, it will confess to anything." – Hal Varian, Computer Mediated Transactions

Learning = Representation + Evaluation + Optimization

It’s Generalization that counts

- The fundamental goal of machine learning is to generalize beyond the examples in the training set

Data alone is not enough

- Induction not deduction – Every learner should embody some knowledge or assumptions beyond the data it is given in order to generalize beyond it

Machine Learning is not magic – one cannot get something from nothing

- In order to infer, one needs the knobs & the dials
- One also needs a rich expressive dataset
Classification - Spark API

- Logistic Regression
- SVMWithSGD
- DecisionTrees
- Data as LabelledPoint (we will see in a moment)
- `DecisionTree.trainClassifier(data, numClasses, categoricalFeaturesInfo, impurity="gini", maxDepth=4, maxBins=100)`
- Impurity – “entropy” or “gini”
- `maxBins` = control to throttle communication at the expense of accuracy
  - Larger = Higher Accuracy
  - Smaller = less communication (as # of bins = number of instances)
- Data Adaptive – i.e. decision tree samples on the driver and figures out the bin spacing i.e. the places you slice for binning
- *Spark = Intelligent Framework - need this for scale*
Lookout for these interesting Spark features

- Concept of Labeled Point & how to create an RDD of LPs
- Print the tree
- Calculate Accuracy & MSE from RDDs
Anatomy Of a Kaggle Competition
Kaggle Data Science Competitions

- Hosts Data Science Competitions

- Competition Attributes:
  - Dataset
  - Train
  - Test (Submission)
  - Final Evaluation Data Set (We don’t see)
  - Rules
  - Time boxed
  - Leaderboard
  - Evaluation function
  - Discussion Forum
  - Private or Public
The Datasets

Titanic Passenger Metadata
- Small
- 3 Predictors
  - Class
  - Sex
  - Age
- Survived?

Walmart Store Forecasting

City Bike Sharing Prediction
(Washington D.C.)
Train.csv
Taken from Titanic Passenger Manifest

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survived</td>
<td>0-No, 1=yes</td>
</tr>
<tr>
<td>Pclass</td>
<td>Passenger Class ( 1st, 2nd, 3rd )</td>
</tr>
<tr>
<td>Sibsp</td>
<td>Number of Siblings/Spouses Aboard</td>
</tr>
<tr>
<td>Parch</td>
<td>Number of Parents/Children Aboard</td>
</tr>
<tr>
<td>Embarked</td>
<td>Port of Embarkation</td>
</tr>
<tr>
<td></td>
<td>C = Cherbourg</td>
</tr>
</tbody>
</table>

Titanic Passenger Metadata
- Small
- 3 Predictors
  - Class
  - Sex
  - Age
- Survived?
## Submission

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>PassengerId</td>
<td>Pclass</td>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
<td>SibSp</td>
<td>Parch</td>
<td>Ticket</td>
<td>Fare</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>892</td>
<td>3</td>
<td>Kelly, Mr. Jar</td>
<td>male</td>
<td>34.5</td>
<td>0</td>
<td>0</td>
<td>330911</td>
<td>7.8292</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>893</td>
<td>3</td>
<td>Wilkes, Mrs. female</td>
<td>47</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>363272</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>894</td>
<td>2</td>
<td>Myles, Mr. Ti</td>
<td>male</td>
<td>62</td>
<td>0</td>
<td>0</td>
<td>240276</td>
<td>9.6875</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>895</td>
<td>3</td>
<td>Wirz, Mr. Alt</td>
<td>male</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>315154</td>
<td>8.6625</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>896</td>
<td>3</td>
<td>Hirvonen, Mrs. female</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>3101298</td>
<td>12.2875</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>897</td>
<td>3</td>
<td>Svensson, Mr. male</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>7538</td>
<td>9.225</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1305</td>
<td>3</td>
<td>Spector, Mr.</td>
<td>male</td>
<td>0</td>
<td>0</td>
<td>A.5. 3236</td>
<td>8.05</td>
<td>S</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1306</td>
<td>1</td>
<td>Oliva y Ocan</td>
<td>female</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>PC 17758</td>
<td>108.9</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1307</td>
<td>3</td>
<td>Saether, Mr.</td>
<td>male</td>
<td>38.5</td>
<td>0</td>
<td>0</td>
<td>SOTON/O.Q.</td>
<td>7.25</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1308</td>
<td>3</td>
<td>Ware, Mr. Fr</td>
<td>male</td>
<td>0</td>
<td>0</td>
<td>359309</td>
<td>8.05</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>1309</td>
<td>3</td>
<td>Peter, Master</td>
<td>male</td>
<td>1</td>
<td>1</td>
<td>2668</td>
<td>22.3583</td>
<td>C</td>
</tr>
</tbody>
</table>

- 418 lines; 1st column should have 0 or 1 in each line
- Evaluation:
  - \% correctly predicted
Data Science “folk knowledge” (Wisdom of Kaggle)
Jeremy’s Axioms

- Iteratively explore data
- Tools
  - Excel Format, Perl, Perl Book, Spark!
- Get your head around data
  - Pivot Table
- Don’t over-complicate
- If people give you data, don’t assume that you need to use all of it
- Look at pictures!
- History of your submissions – keep a tab
- Don’t be afraid to submit simple solutions
  - We will do this during this workshop

Session-2 : Kaggle, Classification & Trees

1. Notebook : 04_Titanic-01
   ① Read Training Data
   ② Henry the Sixth Model
   ③ Submit to Kaggle

2. Notebook : 05_Titanic-02
   ① Decision Tree Model
   ② CE-31 Template
      ① Create Randomforest Model
      ② Predict Testset
      ③ Submit to Kaggle

3. Notebook : 06_Titanic-03
   ① CE-32 Solution
      ① RandomForest Model
      ② Predict Testset
      ③ Submit Solution 2
   ② Discussion about Models
Trees, Forests & Classification

• Discuss Random Forest
  o Boosting, Bagging
  o Data de-correlation

• Why it didn’t do better in Titanic dataset

• Data Science Folk Wisdom
  o http://www.slideshare.net/ksankar/data-science-folk-knowledge
## Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Kaggle Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender based model</td>
<td>~0.7655 Rank : 1276</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>~0.775124, Rank : 1147</td>
</tr>
<tr>
<td>Random Forest</td>
<td>~0.77512</td>
</tr>
</tbody>
</table>

### Gender Based Model

Dick, The butcher to Jack Cade
Dick: The first thing we do, let's kill all the men.
Cade: Nay, that I mean to do.


*Henry the Sixth, Part 2, Act 4, Scene 2*

Why didn’t RF do better? Bias/Variance
See next slide
Why didn’t RF do better? Bias/Variance

- **High Bias**
  - Due to Underfitting
  - Add more features
  - More sophisticated model
    - Quadratic Terms, complex equations, …
  - Decrease regularization

- **High Variance**
  - Due to Overfitting
  - Use fewer features
  - Use more training sample
  - Increase Regularization

'Bias is a learner’s tendency to consistently learn the same wrong thing.' -- Pedro Domingos

Ref: Strata 2013 Tutorial by Olivier Grisel

http://www.slideshare.net/ksankar/data-science-folk-knowledge
## Decision Tree – Best Practices

DecisionTree.trainClassifier(data, numClasses, categoricalFeaturesInfo, impurity="gini", maxDepth=4, maxBins=100)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxDepth</td>
<td>Tune with Data/Model Selection</td>
</tr>
<tr>
<td>maxBins</td>
<td>Set low, monitor communications, increase if needed</td>
</tr>
<tr>
<td># RDD partitions</td>
<td>Set to # of cores</td>
</tr>
<tr>
<td></td>
<td>• Usually the recommendation is that the RDD partitions should be over partitioned ie “more partitions than cores”, because tasks take different times, we need to utilize the compute power and in the end they average out</td>
</tr>
<tr>
<td></td>
<td>• But for Machine Learning especially trees, all tasks are approx equal computationally intensive, so over partitioning doesn’t help</td>
</tr>
<tr>
<td></td>
<td>• Joe Bradley talk (reference below) has interesting insights</td>
</tr>
</tbody>
</table>

https://speakerdeck.com/jkbradley/mllib-decision-trees-at-sf-scala-baml-meetup
Boosting

- “Output of weak classifiers into a powerful committee”
- Final Prediction = weighted majority vote
- Later classifiers get misclassified points
  - With higher weight,
  - So they are forced
  - To concentrate on them
- AdaBoost (*AdaptiveBoosting*)
- Boosting vs Bagging
  - Bagging – independent trees <- Spark shines here
  - Boosting – successively weighted

**Goal**
- Model Complexity (-)
- Variance (-)
- Prediction Accuracy (+)
Random Forests

- Builds large collection of de-correlated trees & averages them
- Improves Bagging by selecting i.i.d* random variables for splitting
- Simpler to train & tune
- “Do remarkably well, with very little tuning required” – ESLII
- Less susceptible to over fitting (than boosting)
- Many RF implementations
  - Original version - Fortran-77 ! By Breiman/Cutler
  - Python, R, Mahout, Weka, Milk (ML toolkit for py), matlab
  - And of course, Spark !

* i.i.d - independent identically distributed
+ http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
Random Forests

• While Boosting splits based on best among *all* variables, RF splits based on best among *randomly chosen* variables
• Simpler because it requires two variables – no. of Predictors (typically $\sqrt{k}$) & no. of trees (500 for large dataset, 150 for smaller)
• Error prediction
  o For each iteration, predict for dataset that is not in the sample (OOB data)
  o Aggregate OOB predictions
  o Calculate Prediction Error for the aggregate, which is basically the OOB estimate of error rate
    • Can use this to search for optimal # of predictors
  o We will see how close this is to the actual error in the Heritage Health Prize
• Assumes equal cost for mis-prediction. Can add a cost function
• Proximity matrix & applications like adding missing data, dropping outliers

Ref: R News Vol 2/3, Dec 2002
Statistical Learning from a Regression Perspective : Berk
A Brief Overview of RF by Dan Steinberg
Ensemble Methods

- **Goal**
  - Model Complexity (-)
  - Variance (-)
  - Prediction Accuracy (+)

- **Two Step**
  - Develop a set of learners
  - Combine the results to develop a composite predictor

- **Ensemble methods can take the form of:**
  - Using different algorithms,
  - Using the same algorithm with different settings
  - Assigning different parts of the dataset to different classifiers

- **Bagging & Random Forests** are examples of ensemble method

Ref: Machine Learning In Action
Deepdive: Leverage parallelism of RDDs, sparse vectors, etc.
Lunch
Back at 1:30
Clustering - Hands On:

• Normalization & Centering
• Clustering
• Optimizing \( k \) based on cohesively of the clusters (WSSE)
Data Science “folk knowledge” (3 of A)

- More Data Beats a Cleverer Algorithm
  - Or conversely select algorithms that improve with data
  - Don’t optimize prematurely without getting more data

- Learn many models, not Just One
  - Ensembles! – Change the hypothesis space
  - Netflix prize
  - E.g. Bagging, Boosting, Stacking

- Simplicity Does not necessarily imply Accuracy

- Representable Does not imply Learnable
  - Just because a function can be represented does not mean it can be learned

- Correlation Does not imply Causation

- http://doubleclix.wordpress.com/2014/03/07/a-glimpse-of-google-nasa-peter-norvig/
- A few useful things to know about machine learning - by Pedro Domingos
  http://dl.acm.org/citation.cfm?id=2347755
Session-3 : Clustering

1. **Notebook : 07_Cluster-1**
   ① Read Data
   ② Cluster
   ③ Modeling Exercise-41-Template

2. **Notebook : 08_Cluster-2**
   ① ME-41-Solution
   ② Center and Scale
   ③ Cluster
   ④ Inspect centroid
   ⑤ CE-42-Template : Cluster Semantics

3. **Notebook : 09_Cluster-3**
   ① CE-42 Solution
   ② Cluster Semantics - Discussion
Clustering - Theory

- Clustering is unsupervised learning
- While the computers can dissect a dataset into “similar” clusters, it still needs human direction & domain knowledge to interpret & guide
- Two types:
  - Centroid based clustering – k-means clustering
  - Tree based Clustering – hierarchical clustering
- Spark implements the Scalable Kmeans++
Lookout for these interesting Spark features

- Application of Statistics toolbox
- Center & Scale RDD
- Filter RDDs
Clustering - API

- from pyspark.mllib.clustering import KMeans
- Kmeans.train
- train(cls, data, k, maxIterations=100, runs=1, initializationMode="k-means||")
- K = number of clusters to create, default=2
- initializationMode = The initialization algorithm. This can be either "random" to choose random points as initial cluster centers, or "k-means||" to use a parallel variant of k-means++ (Bahmani et al., Scalable K-Means++, VLDB 2012). Default: k-means||
- KMeansModel.predict
- Maps a point to a cluster
## Interpretation

### Best Customers, Still lots of non-flying miles

<table>
<thead>
<tr>
<th>C#</th>
<th>AVG</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><img src="image" alt="Table 1" /></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><img src="image" alt="Table 2" /></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><img src="image" alt="Table 3" /></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><img src="image" alt="Table 4" /></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><img src="image" alt="Table 5" /></td>
</tr>
</tbody>
</table>

**Note:**
- This is just a sample interpretation.
- In real life we would “noodle” over the clusters & tweak them to be useful, interpretable and distinguishable.
- May be 3 is more suited to create targeted promotions

- Not that active – Give them flight coupons to encourage them to fly more. Ask them why they are not flying. May be they are flying to destinations (say Jupiter) where InterGallactic has less gates.

- Very active on-line. Why are they coming to us instead of Amazon?
Epilogue

- KMeans in Spark has enough controls
- It does a decent job
- We were able to control the clusters based on our experience (2 cluster is too low, 10 is too high, 5 seems to be right)
- We can see that the Scalable KMeans has control over runs, parallelism et al. (Home work : Explore the scalability)
- We were able to interpret the results with domain knowledge and arrive at a scheme to solve the business opportunity
- Naturally we would tweak the clusters to fit the business viability. 20 clusters with corresponding promotion schemes are unwieldy, even if the WSSE is the minimum.
Recommendation - Hands On:

- Movie Lens - Medium Data
- Movie Lens - Large Data (Homework)
The future of human-machine & Augmented Cognition

- The future is partnership with machines ie let them do what they are best at. I had written about this earlier – we really do not want machines to be like us!
- In that sense augmented cognition is key

And, don’t belong to the B-Ark!
Session-4 : Recommendation at Scale

1. Notebook-10_Reco-1
   ① Read Movielens medium data
   ② CE-51 Template – Partition Data

   ① CE-51 Solution
   ② ALS Slide
   ③ Train ALS & Predict
   ④ Calculate Model Performance
Recommendation & Personalization - Spark

- **Learning Models** - fit parameters as it gets more data
- **Dynamic Models** – model selection based on context
  - Knowledge Based
  - Demographic Based
  - Content Based
  - Collaborative Filtering
    - Item Based
    - User Based
  - Latent Factor based
- **Automated Analytics** - Let Data tell story
  - Feature Learning, AI, Deep Learning

Spark (in 1.1.0) implements the user based ALS collaborative filtering

Ref:
ALS - Collaborative Filtering for Implicit Feedback Datasets, Yifan Hu; AT&T Labs., Florham Park, NJ; Koren, Y.; Volinsky, C.
ALS-WR - Large-Scale Parallel Collaborative Filtering for the Netflix Prize, Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, Rong
Spark Collaborative Filtering API

• ALS.train(cls, ratings, rank, iterations=5, lambda_=0.01, blocks=-1)
• ALS.trainImplicit(cls, ratings, rank, iterations=5, lambda_=0.01, blocks=-1, alpha=0.01)
• MatrixFactorizationModel.predict(self, user, product)
• MatrixFactorizationModel.predictAll(self, usersProducts)
Theory: Matrix Factorization, SVD,...
On-line k-means, spark streaming
Singular Value Decomposition on Spark
Singular Value Decomposition
Singular Value Decomposition

Two cases

» Tall and Skinny

» Short and Fat (not really)

» Roughly Square

SVD method on RowMatrix takes care of which one to call.
Tall and Skinny SVD

- Given $m \times n$ matrix $A$, with $m \gg n$.
- We compute $A^T A$.
- $A^T A$ is $n \times n$, considerably smaller than $A$.
- $A^T A$ is dense.
- Holds dot products between all pairs of columns of $A$.

\[ A = U \Sigma V^T \quad A^T A = V \Sigma^2 V^T \]
Tall and Skinny SVD

\[ A^T A = V \Sigma^2 V^T \] Gets us \( V \) and the singular values

\[ A = U \Sigma V^T \] Gets us \( U \) by one matrix multiplication
Square SVD

ARPACK: Very mature Fortran77 package for computing eigenvalue decompositions

JNI interface available via netlib-java

Distributed using Spark – how?
Square SVD via ARPACK

Only interfaces with distributed matrix via matrix-vector multiplies

\[ K_n = \begin{bmatrix} b & Ab & A^2b & \cdots & A^{n-1}b \end{bmatrix} \]

The result of matrix-vector multiply is small.

The multiplication can be distributed.
### Square SVD

<table>
<thead>
<tr>
<th>Matrix size</th>
<th>Number of nonzeros</th>
<th>Time per iteration (s)</th>
<th>Total time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23,000,000 x 38,000</td>
<td>51,000,000</td>
<td>0.2</td>
<td>10</td>
</tr>
<tr>
<td>63,000,000 x 49,000</td>
<td>440,000,000</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>94,000,000 x 4,000</td>
<td>1,600,000,000</td>
<td>0.5</td>
<td>50</td>
</tr>
</tbody>
</table>

With 68 executors and 8GB memory in each, looking for the top 5 singular vectors.
Communication-Efficient $A^TA$

All pairs similarity on Spark (DIMSUM)
All pairs Similarity

All pairs of cosine scores between n vectors

» Don’t want to brute force \((n \text{ choose } 2)\) \(m\)

» Essentially computes \(A^T A\)

Compute via DIMSUM

» Dimension Independent Similarity
  Computation using MapReduce
Intuition

Sample columns that have many non-zeros with lower probability.

On the flip side, columns that have fewer non-zeros are sampled with higher probability.

Results provably correct and independent of larger dimension, m.
Spark implementation

// Load and parse the data file.
val rows = sc.textFile(filename).map { line =>
    val values = line.split(' ').map(_.toDouble)
    Vectors.dense(values)
}
val mat = new RowMatrix(rows)

// Compute similar columns perfectly, with brute force.
val simsPerfect = mat.columnSimilarities()

// Compute similar columns with estimation using DIMSUM
val simsEstimate = mat.columnSimilarities(threshold)
MLlib + \{Streaming, GraphX, SQL\}
A General Platform

Standard libraries included with Spark

- Spark SQL structured
- Spark Streaming real-time
- GraphX graph
- MLlib machine learning

Spark Core
Benefit for Users

Same engine performs data extraction, model training and interactive queries.
MLlib + Streaming

As of Spark 1.1, you can train linear models in a streaming fashion, k-means as of 1.2

Model weights are updated via SGD, thus amenable to streaming

More work needed for decision trees
MLlib + SQL

df = context.sql("select latitude, longitude from tweets")
model = pipeline.fit(df)

DataFrames in Spark 1.3! (March 2015)
Powerful coupled with new pipeline API
MLlib + GraphX

// assemble link graph
val graph = Graph(pages, links)
val pageRank: RDD[(Long, Double)] = graph.staticPageRank(10).vertices

// load page labels (spam or not) and content features
val labelAndFeatures: RDD[(Long, (Double, Seq[Int, Double]))] = ...
val training: RDD[LabeledPoint] =
  labelAndFeatures.join(pageRank).map {
    case (id, ((label, features), pageRank)) =>
      LabeledPoint(label, Vectors.sparse(features ++ (1000, pageRank))
  }

// train a spam detector using logistic regression
val model = LogisticRegressionWithSGD.train(training)
Future of MLlib
Goals for next version

Tighter integration with DataFrame and spark.ml API

Accelerated gradient methods & Optimization interface

Model export: PMML (current export exists in Spark 1.3, but not PMML, which lacks distributed models)

Scaling: Model scaling (e.g. via Parameter Servers)
Research Goal: General Distributed Optimization

Distribute CVX by backing CVXPY with PySpark

Easy-to-express distributable convex programs

Need to know less math to optimize complicated objectives

```python
from cvxpy import *

# Create two scalar optimization variables.
x = Variable()
y = Variable()

# Create two constraints.
constraints = [x + y == 1,
              x - y >= 1]

# Form objective.
obj = Minimize(square(x - y))

# Form and solve problem.
prob = Problem(obj, constraints)
prob.solve()  # Returns the optimal value.
print "status:", prob.status
print "optimal value", prob.value
print "optimal var", x.value, y.value

status: optimal
optimal value 0.99999999323
optimal var 0.999999998248 1.75244914951e-09
```
Spark Community

Most active open source community in big data

200+ developers, 50+ companies contributing

Contributors in past year
Continuing Growth

source: ohloh.net

Contributors per month to Spark
Spark and ML

Spark has all its roots in research, so we hope to keep incorporating new ideas!
Coffee-Break
Back at 3:15
Hands On:

• Mood Of the Union
• RecSys 2015 Challenge
The Art of ELO Ranking & Super Bowl XLIX

○ The real formula is

\[ E_a = \frac{1}{1 + 10(R_b - R_a)/400} = \frac{1}{1 + 10(R_a - R_b)/400} \]

○ Not what is written on the glass!

\[ E_a = \frac{1}{1 + 10(R_b - R_a)/400} \]

○ But then that is Hollywood!

I need the Algorithm, I need the Algorithm

- Mark Z to Eduardo S

Eduardo Saverin: Hey, Mark.
Mark Zuckerberg: Wardo.
Eduardo Saverin: You and Erica split up.
Mark Zuckerberg: [confused] How did you know that?
Eduardo Saverin: It's on your blog.
Mark Zuckerberg: Yeah.
Eduardo Saverin: Are you all right?
Mark Zuckerberg: I need you.
Eduardo Saverin: I'm here for you.
Mark Zuckerberg: No, I need the algorithm you used to rank chess players.
Eduardo Saverin: Are you OK?

Session-5: Mood of the Union – Data Science on SOTU by POTUS

1. DataScience/12_SOTU-1
   ① Read BO
   ② CE-61 Template: Has BO changed since 2014?

2. DataScience/13_SOTU-2
   ① CE-61 Solution
   ② Read GW
   ③ Preprocess
   ④ CE-62 Template: What mood the country was in 1790-1796 vs. 2009-2015

3. Notebook: 14_SOTU-3
   ① CE-62 Solution
   ② Homework
      ① GWB vs Clinton
      ② WJC vs AL
   ③ Discussions
Epilogue

• Interesting Exercise

• Highlights
  o Map-reduce in a couple of lines !
    • But it is not exactly the same as Hadoop Mapreduce (see the excellent blog by Sean Owen¹)
  o Set differences using substractByKey
  o Ability to sort a map by values (or any arbitrary function, for that matter)

• To Explore as homework:
  o TF-IDF in
    http://spark.apache.org/docs/latest/mllib-feature-extraction.html#tf-idf

¹ http://blog.cloudera.com/blog/2014/09/how-to-translate-from-mapreduce-to-apache-spark/
Session-7 : Predict Buying Pattern
Recsys 2015 Challenge

   1. Read Data
   2. Explore Options
   3. CE-71
After Hours

Homework
Notebooks to Explore

• Run thru all the homework in your Databricks cloud

```bash
> _99_RecSys-2015-01
> _HW1_SQL101
> _HW2_SQL102
> _HW3_Titanic-Naive Ba...
> _HW4-Convert_SOTU_T...
> _HW5_GraphX
> _MatrixComps
```
GraphX examples
GraphX:

spark.apache.org/docs/latest/graphx-programming-guide.html

Key Points:

- graph-parallel systems
- importance of workflows
- optimizations
**GraphX: Further Reading…**

*PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs*
J. Gonzalez, Y. Low, H. Gu, D. Bickson, C. Guestrin

*Pregel: Large-scale graph computing at Google*
Grzegorz Czajkowski, et al.
[googleresearch.blogspot.com/2009/06/large-scale-graph-computing-at-google.html](googleresearch.blogspot.com/2009/06/large-scale-graph-computing-at-google.html)

*GraphX: Unified Graph Analytics on Spark*
Ankur Dave, Databricks
[databricks-training.s3.amazonaws.com/slides/graphx@sparksummit_2014-07.pdf](databricks-training.s3.amazonaws.com/slides/graphx@sparksummit_2014-07.pdf)

*Advanced Exercises: GraphX*
**GraphX: Example – simple traversals**


```scala
import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD

case class Peep(name: String, age: Int)

val nodeArray = Array(
  (1L, Peep("Kim", 23)), (2L, Peep("Pat", 31)),
  (3L, Peep("Chris", 52)), (4L, Peep("Kelly", 39)),
  (5L, Peep("Leslie", 45))
)

val edgeArray = Array(
  Edge(2L, 1L, 7),
  Edge(2L, 4L, 2),
  Edge(3L, 2L, 4),
  Edge(3L, 5L, 3),
  Edge(4L, 1L, 1),
  Edge(5L, 3L, 9)
)

val nodeRDD: RDD[(Long, Peep)]= sc.parallelize(nodeArray)
val edgeRDD: RDD[Edge[Int]]= sc.parallelize(edgeArray)

val g: Graph[Peep, Int]= Graph(nodeRDD, edgeRDD)

val results = g.triplets.filter(t => t.attr > 7)

for (triplet <- results.collect)
  println(s"${triplet.srcAttr.name} loves ${triplet.dstAttr.name}"")
```

84 GraphX: Example – simple traversals
**GraphX: Example – routing problems**

What is the cost to reach **node 0** from any other node in the graph? This is a common use case for graph algorithms, e.g., **Dijkstra**
GraphX: Coding Exercise

Run /<your folder>/_DataSCience/08.graphx in your folder:
Case Studies
Case Studies: Apache Spark, DBC, etc.

Additional details about production deployments for Apache Spark can be found at:

https://cwiki.apache.org/confluence/display/SPARK/Powered+By+Spark

https://databricks.com/blog/category/company/partners

http://go.databricks.com/customer-case-studies
Case Studies: Automatic Labs

Spark Plugs Into Your Car
Rob Ferguson
spark-summit.org/east/2015/talk/spark-plugs-into-your-car

Automatic creates personalized driving habit dashboards

• wanted to use Spark while minimizing investment in DevOps
• provides data access to non-technical analysts via SQL
• replaced Redshift and disparate ML tools with single platform
• leveraged built-in visualization capabilities in notebooks to generate dashboards easily and quickly
Case Studies: Twitter

Spark at Twitter: Evaluation & Lessons Learnt
Sriram Krishnan
slideshare.net/krishflix/seattle-spark-meetup-spark-at-twitter

• Spark can be more interactive, efficient than MR
  • support for iterative algorithms and caching
  • more generic than traditional MapReduce

• Why is Spark faster than Hadoop MapReduce?
  • fewer I/O synchronization barriers
  • less expensive shuffle
  • the more complex the DAG, the greater the performance improvement
Pearson uses Spark Streaming for next generation adaptive learning platform

Dibyendu Bhattacharya

databricks.com/blog/2014/12/08/pearson-uses-spark-streaming-for-next-generation-adaptive-learning-platform.html

- Kafka + Spark + Cassandra + Blur, on AWS on a YARN cluster
- single platform/common API was a key reason to replace Storm with Spark Streaming
- custom Kafka Consumer for Spark Streaming, using Low Level Kafka Consumer APIs
- handles: Kafka node failures, receiver failures, leader changes, committed offset in ZK, tunable data rate throughput
Case Studies: Concur

Unlocking Your Hadoop Data with Apache Spark and CDH5
Denny Lee
slideshare.net/Concur/unlocking-your-hadoop-data-with-apache-spark-and-cdh5

- leading provider of spend management solutions and services
- delivers recommendations based on business users’ travel and expenses – “to help deliver the perfect trip”
- use of traditional BI tools with Spark SQL allowed analysts to make sense of the data without becoming programmers
- needed the ability to transition quickly between Machine Learning (MLLib), Graph (GraphX), and SQL usage
- needed to deliver recommendations in real-time
Case Studies:  Stratio

Stratio Streaming: a new approach to Spark Streaming
David Morales, Oscar Mendez
spark-summit.org/2014/talk/stratio-streaming-a-new-approach-to-spark-streaming

- Stratio Streaming is the union of a real-time messaging bus with a complex event processing engine atop Spark Streaming
- allows the creation of streams and queries on the fly
- paired with Siddhi CEP engine and Apache Kafka
- added global features to the engine such as auditing and statistics
Case Studies: Spotify

Collaborative Filtering with Spark

Chris Johnson

slidesshare.net/MrChrisJohnson/collaborative-filtering-with-spark

- collab filter (ALS) for music recommendation
- Hadoop suffers from I/O overhead
- show a progression of code rewrites, converting a Hadoop-based app into efficient use of Spark
Guavus Embeds Apache Spark into its Operational Intelligence Platform
Deployed at the World’s Largest Telcos

**Eric Carr**


- 4 of 5 top mobile network operators, 3 of 5 top Internet backbone providers, 80% MSOs in NorAm
- analyzing 50% of US mobile data traffic, +2.5 PB/day
- latency is critical for resolving operational issues before they cascade: 2.5 MM transactions per second
- “analyze first” not “store first ask questions later”
Case Studies: Radius Intelligence

From Hadoop to Spark in 4 months, Lessons Learned
Alexis Roos
http://youtu.be/o3-lokJFqYc

- building a full SMB index took 12+ hours using Hadoop and Cascading
- pipeline was difficult to modify/enhance
- Spark increased pipeline performance 10x
- interactive shell and notebooks enabled data scientists to experiment and develop code faster
- PMs and business development staff can use SQL to query large data sets
Further Resources + Q&A

Scalable Machine Learning
Learn the underlying principles required to develop scalable machine learning pipelines and gain hands-on experience using Apache Spark.
community:

spark.apache.org/community.html

events worldwide: goo.gl/2YqJZK

video+preso archives: spark-summit.org

resources: databricks.com/spark-training-resources

workshops: databricks.com/spark-training
We enjoyed a lot preparing the materials...
Hope you enjoyed more attending...

Do. Or do not. There is no try.

#1 MISTAKE

TIME MANAGEMENT

Checking \+ social media

SO THEN

1. Write tomorrow's top priorities on a Post-it
2. X-off the bottom 5
3. Stick the sticky on your computer
4. Block 30 min to work on your top priority
5. Before checking email, Facebook, or Twitter, write down what you are about to do

The Disciplined Pursuit: Less

THANK YOU

Grazie