Deep Analytics Pipeline
A Benchmark Proposal

Milind Bhandarkar
Chief Scientist, Pivotal Software
(Twitter: @techmilind)
Quest for “Typical” Workloads

- Benchmarks most relevant if representative
- Tune systems for broadly applicable workloads
- Designing optimized systems: Make common tasks fast, other tasks possible
Encouraging Early Results

- Analyzed characteristics of 1M+ real Hadoop jobs on production clusters at Yahoo, 100+ features
- Identified 8 Job types
- Verified with GridMix 3

Big Data Benchmark Community

• Formed in January 2012 under the Center for Large Data Systems Umbrella (clds.sdsc.edu)

• 200+ members on the mailing list

• Workshops on Big Data Benchmarking (San Jose, 2012, Pune, India 2012, Xian, China 2013, San Jose CA 2013, Germany 2014)
Benchmark Proposals

• BigBench: Rabl et al, Published in SIGMOD 2013, Full Specification published in WBDB proceedings (Springer-Verlag)
• Data Model enhanced from TPC-DS
• Added Text Mining, Relevance
• TPC-BD: Formed September 2013, Terasort + TPC-rigor
BigData Top 100

- Modeled after Top500 & Graph500 in HPC Community
- Proposal Presented at Strata Conference in February 2013
- Workload: BigBench (most likely)
Drivers for Big Data

- Ubiquitous Connectivity
- Sensors Everywhere
- Democratization of Content
Big Data Sources

• Events
  • Direct - Human Initiated
  • Indirect - Machine Initiated

• Software Sensors (Clickstreams, Locations)

• Public Content (blogs, tweets, Status updates, images, videos)
Online: Major Data Sources

- Search Queries
- User
- Advertisements (Display, Search)
- Content (Web Pages, Blogs, News Articles, Media)
“User” Modeling

- Objective: Determine User-Interests by mining user-activities
- Large dimensionality of possible user activities
- Typical user has sparse activity vector
- Event attributes change over time
User-Modeling Pipeline

- Data Acquisition, Normalization, Sessionization
- Feature and Target Generation
- Model Training
- Offline Scoring & Evaluation
- Batch Scoring & Upload to serving
# Data Acquisition

<table>
<thead>
<tr>
<th>User</th>
<th>Time</th>
<th>Event</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0</td>
<td>T0</td>
<td>Visited Auto website</td>
<td>Web Server logs</td>
</tr>
<tr>
<td>U0</td>
<td>T1</td>
<td>Searched for “Car Insurance”</td>
<td>Search Logs</td>
</tr>
<tr>
<td>U0</td>
<td>T2</td>
<td>Browsed stock quotes</td>
<td>Web Server Logs</td>
</tr>
<tr>
<td>U0</td>
<td>T3</td>
<td>Saw ad for “discount brokerage”, did not click</td>
<td>Ad Logs</td>
</tr>
<tr>
<td>U0</td>
<td>T4</td>
<td>Checked Mail</td>
<td>Web Server Logs</td>
</tr>
<tr>
<td>U0</td>
<td>T5</td>
<td>Clicked Ad for “Auto Insurance”</td>
<td>Ad Logs, Click Logs</td>
</tr>
</tbody>
</table>
## Normalization

<table>
<thead>
<tr>
<th>User</th>
<th>Time</th>
<th>Event</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0</td>
<td>T0</td>
<td>View</td>
<td>Category: Autos, Tag: Mercedes Benz</td>
</tr>
<tr>
<td>U0</td>
<td>T1</td>
<td>Query</td>
<td>Category: Insurance, Tag: Auto</td>
</tr>
<tr>
<td>U0</td>
<td>T2</td>
<td>View</td>
<td>Category: Finance, Tag: EMC</td>
</tr>
<tr>
<td>U0</td>
<td>T3</td>
<td>View-Click</td>
<td>Category: Finance, Tag: Brokerage</td>
</tr>
<tr>
<td>U0</td>
<td>T4</td>
<td>Browse</td>
<td>Irrelevant Event, Dropped</td>
</tr>
<tr>
<td>U0</td>
<td>T5</td>
<td>View+Click</td>
<td>Category: Insurance, Tag: Auto</td>
</tr>
</tbody>
</table>
Features & Targets
Targets

• User-Actions of Interest
  • Clicks on Ads & Content
  • Site & Page visits
  • Conversion Events
    • Purchases, Quote requests
    • Sign-Up for membership etc
Features

• Summary of user activities over a time-window

• Aggregates, moving averages, rates over various time-windows

• Incrementally updated
Joining Targets & Features

• Target rates very low: 0.01% ~ 1%
• First, construct targets
• Filter user activity without targets
• Join feature vector with targets
Model Training

- Regressions
- Boosted Decision Trees
- Naive Bayes
- Support Vector Machines
- Maximum Entropy modeling
- Constrained Random Fields
Offline Scoring & Evaluation

- Apply model weights to features
- Pleasantly parallel
- Sort by scores and compute metrics
- Evaluate metrics
Batch Scoring

- Apply models to features from all user activity
- Upload scores to serving systems
Issues

• Different modeling techniques for different kinds of data
• Different notions of a “session”
• Widely varying number of events per entity
Proposal: 5 Classes

- Tiny (100K entities, 10 events per entity)
- Small (1M entities, 10 events per entity)
- Medium (10M entities, 100 events per entity)
- Large (100M entities, 1000 events per entity)
- Huge (1B entities, 1000 events per entity)
Proposal: Publish results for every stage

- Data pipelines constructed by mix-and-match of various stages
- Different modeling techniques per class
- Need to publish performance numbers for every stage
Questions ?