Big data management and predictive analytics for customer transactions & operations using Apache Spark and AWS

Large Scale Distributed Machine Learning

BDA761 Big Data Mgmt in a Supercomputing environment

MUDIT UPPAL
Problem with Big Data(s)

- Machine learning practices at scale for PB/TB data
- A framework which provides and computes models using virtual nodes with processors and memory getting cheaper every year
- Using GPU + multi-threading + make use of multiple cores
- Goal: Thinking in ‘big data’; create a tool which can be used in any operations/Sales/customer analysis
- Parallelizing — DATA and MODEL
Case study

❖ *Rossmann*: operates over 3000 drug stores in 7 european countries

❖ Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality.

❖ DECISION TASKS:

❖ Forecast Sales for upto 6 weeks using stores data, customers, promotion data et cetera

❖ Predicting Sales of ~1000 stores daily

Data Source: Rossmann, Walmart via Kagle
~2.8 million rows data points

**Label:** Sales

**Features:** Store, Sales, customers, Open, date, stateholiday, schoolHoliday, storetype, Assortment, competitionDistance, CompetitionSinceMonth, sinceYear, Promo, Promo2, PromoInterval, Promo2SinceMonth, PromoSince year, DayOfWeek, etc…

<table>
<thead>
<tr>
<th>Id</th>
<th>Store</th>
<th>DayOfWeek</th>
<th>Date</th>
<th>Open</th>
<th>Promo</th>
<th>StateHoliday</th>
<th>SchoolHoliday</th>
<th>Sales</th>
<th>Customers</th>
<th>Open</th>
<th>Promo</th>
<th>StateHoliday</th>
<th>SchoolHoliday</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2015-09-17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5263</td>
<td>555</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2015-09-17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6064</td>
<td>625</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2015-09-17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8314</td>
<td>821</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TripType</th>
<th>VisitNumber</th>
<th>Weekday</th>
<th>Upc</th>
<th>ScanCount</th>
<th>Department/Description</th>
<th>FinelineNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>999</td>
<td>5</td>
<td>Friday</td>
<td>68113155299</td>
<td>1</td>
<td>FINANCIAL SERVICES</td>
<td>1000</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>Friday</td>
<td>60538815800</td>
<td>1</td>
<td>SHOES</td>
<td>8931</td>
</tr>
<tr>
<td>7410811089</td>
<td>1</td>
<td>PERSONAL CARE</td>
<td>4504</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2238403510</td>
<td>2</td>
<td>PAINT AND ACCESSORIES</td>
<td>3565</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2065612364</td>
<td>4</td>
<td>PAINT AND ACCESSORIES</td>
<td>1017</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2065612363</td>
<td>8</td>
<td>PAINT AND ACCESSORIES</td>
<td>1017</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7004545737</td>
<td>12</td>
<td>PAINT AND ACCESSORIES</td>
<td>2802</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2338495318</td>
<td>16</td>
<td>PAINT AND ACCESSORIES</td>
<td>4501</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2338495306</td>
<td>20</td>
<td>PAINT AND ACCESSORIES</td>
<td>3065</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5200010029</td>
<td>30</td>
<td>DSD GROCERY</td>
<td>4506</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9979002561</td>
<td>40</td>
<td>PAINT AND ACCESSORIES</td>
<td>3504</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2206500000</td>
<td>60</td>
<td>MEAT - FRESH &amp; FROZEN</td>
<td>6009</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2238565452</td>
<td>100</td>
<td>PAINT AND ACCESSORIES</td>
<td>1305</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8867930056</td>
<td>150</td>
<td>PAINT AND ACCESSORIES</td>
<td>3504</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2238495200</td>
<td>350</td>
<td>PAINT AND ACCESSORIES</td>
<td>3565</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3012394023</td>
<td>2000</td>
<td>PAINT AND ACCESSORIES</td>
<td>2801</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2745040840</td>
<td>7000</td>
<td>PAINT AND ACCESSORIES</td>
<td>1028</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2545150000</td>
<td>2000</td>
<td>DAIRY</td>
<td>1305</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2311001077</td>
<td>3000</td>
<td>PETS AND SUPPLIES</td>
<td>3300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2745040700</td>
<td>1000</td>
<td>PAINT AND ACCESSORIES</td>
<td>1018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7874204967</td>
<td>5000</td>
<td>HOUSEHOLD CHEMICALS/SUPP</td>
<td>707</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5114139038</td>
<td>2000</td>
<td>PAINT AND ACCESSORIES</td>
<td>4415</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5114139761</td>
<td>3000</td>
<td>PAINT AND ACCESSORIES</td>
<td>4415</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3270011053</td>
<td>4000</td>
<td>PETS AND SUPPLIES</td>
<td>1001</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Store</th>
<th>StoreType</th>
<th>Assortment</th>
<th>CompetitionDistance</th>
<th>CompetitionOpenSinceMonth</th>
<th>CompetitionOpenSinceYear</th>
<th>Promo2</th>
<th>Promo2SinceMonth</th>
<th>Promo2SinceYear</th>
<th>PromoInterval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c</td>
<td>a</td>
<td>1270</td>
<td>9</td>
<td>2008</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>a</td>
<td>570</td>
<td>11</td>
<td>2007</td>
<td>1</td>
<td>13</td>
<td>2010</td>
<td>Jan, Apr, Jul, Oct</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>14130</td>
<td>12</td>
<td>2005</td>
<td>1</td>
<td>14</td>
<td>2011</td>
<td>Jan, Apr, Jul, Oct</td>
</tr>
</tbody>
</table>
Goal

- Aim is to create a Predictive Analytics Framework
  - distributed machine learning for any size dataset
- 3 Demos in this presentation
  - Apache Spark
  - Exploratory DA with R
  - xgBoost (python) ML

Data Source: Rossmann, Walmart via Kaggle
Main Demos

- Exploratory data analysis
- **Apache Spark** (sql spark context) - Distributed ML across multiple machines/nodes
- Linear Regression analysis
- Gradient descent
- Ensemble and boosting algorithms (**XGBoost**)
Tools

- R + python for exploratory analysis
- Apache Spark for implementation
- Hadoop
- Spark MLlib
- AWS cloud (m4 large instances)
- Ganglia (distributed monitoring system to work with clusters)
- pyspark
Data Pipeline

Obtain Data → Split Data → Feature extraction → Find relationships → Supervised learning

Evaluate → Predict

Parse Initial dataset

Use ‘LabeledPoint’ class

Visualize features

Shift labels (starting from zero)

Create and Evaluate Baseline model

Train (via gradient descent) and evaluate a linear regression

Use weights to make predictions

Spark
Demo 1: exploratory analysis

- R+python
- Data and source code can be downloaded from: http://muppal.com
ApacheSpark <——> xgboost

- Spark excels at distributing operations across a cluster while abstracting away many of the underlying implementation details.
- Thinking in terms of RDD(transformations and actions)
- Still under development

**XGBoost:**
- start off with a rough prediction and then building a series of decision trees; with each trying to correct the prediction error of the one before
- Large-scale and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library, on single node, hadoop yarn etc
- XGBoost can also be distributed and scale to Terascale data
- You can define threads with “nthreads =..”
- [https://github.com/dmlc/xgboost](https://github.com/dmlc/xgboost)
\[
\begin{bmatrix}
9 & 3 & 5 \\
4 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
1 & 2 \\
3 & -5 \\
2 & 3
\end{bmatrix}
= 
\begin{bmatrix}
28 & 18 \\
11 & 9
\end{bmatrix}
\]

\[
\begin{bmatrix}
9 & 18 \\
4 & 8
\end{bmatrix}
+ 
\begin{bmatrix}
9 & -15 \\
3 & -5
\end{bmatrix}
+ 
\begin{bmatrix}
10 & 15 \\
4 & 6
\end{bmatrix}
\]

\[
X^T X = \sum_{i=1}^{n} x^{(i)} x^{(i)T} =
\]

Example: \( n = 6; 3 \) workers

workers:

map:

reduce:

\( \left( \sum x^{(i)} \right)^{-1} \)
Data pipeline/Tools

Worker1 - ML Algorithm 1 & 2
Worker2 - ML Algorithm 3 & 4
Worker3 - ML Algorithm 5 & 6
Worker4 - ML Algorithm 7 & 8

Distributed ML Analysis

hdfs
Driver

Spark Driver and Workers

- A Spark program is two programs:
  - A driver program and a workers program
- Worker programs run on cluster nodes or in local threads
- RDDs are distributed across workers

Your application (driver program)
SparkContext
Cluster manager
Local threads
Worker Spark executor
Worker Spark executor
Amazon S3, HDFS, or other storage
Demo 2/3: xgboost + ML Spark on clusters

- xgboost + Apache Spark/pyspark
- Findings: train-rmspe:0.103954  eval-rmspe:0.094526
- Validating — RMSPE: 0.094526
- Data and source code can be downloaded from: http://muppal.com
Thank you!
References

- https://www.kaggle.com/c/walmart-recruiting-trip-type-classification