Build Recommender Systems, Detect Network Intrusion, and Integrate Deep Learning with Graph Technologies

Zhe Wu  
Architect  
Oracle

Chris Nicholson  
CEO  
Skymind

Charlie Berger  
Senior Director  
Oracle

BIWA 2017
Outline

• Overview of graph technologies
  – Property graph data model
  – Typical use cases of property graph

• Oracle Advanced Analytics and its integration with graph

• Overview of Artificial Intelligence and Deep Learning
  – How can Deep Learning be used together with Graph technologies
  – Case study on network intrusion detection

• Summary and Future Work
Relational Model vs. Graph Model

• Relational Model

• Graph Model

Courtesy: Tom Sawyer 2016
Two Graph Data Models: RDF and Property Graph

**RDF Data Model**
- Data federation
- Knowledge representation
- Inferencing

**Industry Domain**
- Life Sciences
- Health Care
- Publishing
- Finance

**Application Area**
- Social Network Analysis

**Property Graph Model**
- Graph Search & Analysis
- Big Data analytics
- Entity analytics

**Industry Domain**
- National Intelligence
- Public Safety
- Social Media search
- Marketing - Sentiment

---

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
The Property Graph Data Model

- A set of vertices (or nodes)
  - each vertex has a unique identifier.
  - each vertex has a set of in/out edges.
  - each vertex has a collection of key-value properties.

- A set of edges (or links)
  - each edge has a unique identifier.
  - each edge has a head/tail vertex.
  - each edge has a label denoting type of relationship between two vertices.
  - each edge has a collection of key-value properties.

https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model
How is graph analysis important to business?

• What patterns are there in fraudulent behavior?
• Which supplier am I most dependent upon?
• Who is the most influential customer?
• Do my products appeal to certain communities?
• What targeted products or services do I recommend to customers?
Graph Use Case Scenarios

• Fraud detection
  – Find parties in insurance data who are on both sides of multiple claims, who live near each other

• Internet of Things
  – Manage graph of interconnected devices and predict the effect of disruptions across network

• Cyber Security
  – Find entry points and affected machines

• Border Control
  – Analyze flight histories of a suspicious passenger. Identify his co-travelers, co-traveler’s co-travelers, ...
Graph Analysis in Business

**Product Recommendation**
Recommend the most similar item purchased by similar people

**Influencer Identification**
Find out people that are central in the given network – e.g. influencer marketing

**Community Detection**
Identify group of people that are close to each other – e.g. target group marketing

**Graph Pattern Matching**
Find out all the sets of entities that match to the given pattern – e.g. fraud detection
Building a Recommender System
-- with Oracle Big Data Spatial and Graph Property Graph

• Environment
  – **Oracle Big Data Lite** VM 4.5.0+
  – Oracle Big Data Spatial and Graph v1.2.0+
  – SolrCloud 4.10.x

• A “**user-item**” property graph
  – Vertices (items, descriptions, and users)
  – Edges (linking users and items)

Recommendation: *you may also like*
Building a Recommender System
-- with Oracle Big Data Spatial and Graph Property Graph

• BDSG offers multiple approaches and they can be mixed together

**Content-based filtering**
- Match item description
- Match user profile
- Relevancy ranking

**Collaborative filtering**
- People liked similar items in the past will like similar items in the future

**Personalized Page Ranking**
- Randomly navigate from a user to a product, then back to a user, ...
- Randomly jump to starting point(s)

\[ A \rightarrow u \]
\[ u \rightarrow B \]
\[ B \rightarrow w \]
\[ w \rightarrow C \]
\[ \ldots \]
Architecture of **Existing** Property Graph Support

**Graph Analytics**
- Parallel In-Memory Graph Analytics (PGX)

**Data Access Layer**
- Apache Blueprints & Lucene/SolrCloud

**Oracle Spatial and Graph**
- Oracle Database 12.2

**Oracle Big Data Spatial and Graph**
- Apache HBase
- Oracle NoSQL Database

**Property graph formats supported**
- GraphML
- GML
- Graph-SON
- Flat Files
- CSV
- Relational Data Sources

Java APIs
- Java, Groovy, Python, …

Java APIs/JDBC/SQL/PLSQL

REST/Web Service

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Data Access (APIs)

• Blueprints 2.3.0, Gremlin 2.3.0, Rexster 2.3.0
• Groovy shell for accessing property graph data
• REST APIs (through Rexster integration)
• PGQL (Property Graph Query Language)
Text Search through Apache Lucene/SolrCloud

- Integration with Apache Lucene & SolrCloud
- Support manual and auto indexing of Graph elements
  - Manual index:
    - `oraclePropertyGraph.createIndex("my_index", Vertex.class);`
    - `indexVertices = oraclePropertyGraph.getIndex("my_index", Vertex.class);`
    - `indexVertices.put("key", "value", myVertex);`
  - Auto Index
    - `oraclePropertyGraph.createKeyIndex("name", Edge.class);`
    - `oraclePropertyGraph.getEdges("name", "*hello*world");`
  - Enables queries to use syntax like "*oracle* or *graph*"
Support for Cytoscape Open Source Visualization
Integration with Tom Sawyer Perspectives via property graph REST APIs
Oracle Spatial and Graph Property Graph
In-Memory Analyst Performance
Oracle’s In-Memory Analyst vs Spark GraphX 1.1

In-Memory Analyst on 1 node is up to 2 orders of magnitude faster than Spark GraphX distributed execution on 2 to 16 nodes.
In-Memory Analyst on a single machine is 3x – 10x faster than a GraphLab 16-machine distributed execution
Performance on Oracle Database

- **Performance Evaluation (1/2 Rack Exadata)**
  - Loading of PG data from flat files into Oracle Database
    - Twitter graph loaded in 882 seconds (~1.4 billion edges + 4 indexes)
    - Yahoooweb graph loaded in 2468 seconds (~3 billion edges + 4 indexes)
  - Parallel Lucene Indexing
    - Twitter graph data text indexed in 2.6 hrs
    - Yahoooweb graph data text indexed in 5.6 hrs
  - SQL-Based Graph Analytics

<table>
<thead>
<tr>
<th>Graph/Operations</th>
<th>Community Detection</th>
<th>Page Ranking (per iter.)</th>
<th>Triangle Counting</th>
<th>Triangle Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Graph (1.4 billion edges)</td>
<td>3min 10s</td>
<td>70s</td>
<td>69min (34.8 billion triangles)</td>
<td>85s (~35.6 billion triangles)</td>
</tr>
<tr>
<td>YahooWeb (2.9 billion edges)</td>
<td>10min 17s</td>
<td>140s</td>
<td>131min (363.7 billion triangles)</td>
<td>106s (~354 billion triangles)</td>
</tr>
</tbody>
</table>
Oracle Advanced Analytics
Google “Oracle Advanced Analytics”

Oracle Advanced Analytics 12c delivers parallelized in-database implementations of data mining algorithms and integration with open source R. Data analysts use Oracle Data Miner GUI and R to build and evaluate predictive models and leverage R packages and graphs. Application developers deploy Oracle Advanced Analytics models using SQL data mining functions and R. With the Oracle Advanced Analytics option, Oracle extends the Oracle Database to an in-memory analytical form that mines more data and data types, eliminates data movement, and preserves security to anticipate customer behavior, detect patterns, and deliver actionable insights.

Oracle Big Data SQL adds new big data sources and Oracle R Advanced Analytics for Hadoop provides algorithms that run on Hadoop.

Oracle R Advanced Analytics for Hadoop

NEW ORA/R 2.7.0: Introducing the fastest QLM and LM algorithms on Spark with full summary, enhanced Deep Neural Networks and support for Spark MLlib Gaussian Mixture Models.

The latest release of Oracle R Advanced Analytics for Hadoop (ORA/R), version 2.7.0 is one of the components of the Oracle Big Data Connectors software suite. An update to the Oracle Big Data Appliance at its core, ORA/R provides an R interface for manipulating data stored in HDFS, using both R/E transparency capabilities and mapping R/DSO so-routed HDFS data to MySQL, algorithms that can run as MapReduce jobs or inside an Apache Spark container.
### Oracle Advanced Analytics DB Option

**In-Database Machine Learning Algorithms**—SQL & GUI Access

<table>
<thead>
<tr>
<th>Classification</th>
<th>Clustering</th>
<th>Predictive Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Decision Tree</td>
<td>• Hierarchical k-Means</td>
<td>• Clustering</td>
</tr>
<tr>
<td>• Logistic Regression (GLM)</td>
<td>• Orthogonal Partitioning Clustering</td>
<td>• Regression</td>
</tr>
<tr>
<td>• Naïve Bayes</td>
<td>• Expectation-Maximization</td>
<td>• Anomaly Detection</td>
</tr>
<tr>
<td>• Support Vector Machine (SVM)</td>
<td></td>
<td>• Feature Extraction</td>
</tr>
<tr>
<td>• Random Forest</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Regression</th>
<th>Attribute Importance</th>
<th>Feature Extraction &amp; Creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multiple Regression (GLM)</td>
<td>• Minimum Description Length</td>
<td>• Nonnegative Matrix Factorization</td>
</tr>
<tr>
<td>• Support Vector Machine (SVM)</td>
<td>• Unsupervised pair-wise KL div.</td>
<td>• Principal Component Analysis</td>
</tr>
<tr>
<td>• Stepwise Linear Regression</td>
<td></td>
<td>• Singular Value Decomposition</td>
</tr>
<tr>
<td>• Linear Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Generalized Linear Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Multi-Layer Neural Networks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anomaly Detection</th>
<th>Market Basket Analysis</th>
<th>Text Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1-Class Support Vector Machine</td>
<td>• Apriori – Association Rules</td>
<td>• All OAA/ODM SQL ML support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Explicit Semantic Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Time Series</th>
<th>Open Source R Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Single &amp; Double Exp. Smoothing</td>
<td>• Ability to run any R package (9,000+) via Embedded R mode</td>
</tr>
</tbody>
</table>

+ Ability to Mine Unstructured, Structured & Transactional data
+ Partitioned Models
Oracle’s Advanced Analytics (Machine Learning Platform)

Multiple interfaces across platforms — SQL, R, GUI, Dashboards, Apps

**Information Producers**
- SQL Developer/Oracle Data Miner
- OBIEE
- Oracle Cloud

**Information Consumers**
- R Client
- Applications

**Users**
- R programmers
- Data & Business Analysts
- Business Analysts/Mgrs
- Domain End Users

**Platform**
- Hadoop
  - ORAAH
  - Parallel, distributed algorithms
- Oracle Database Enterprise Edition
  - Oracle Advanced Analytics - Database Option
  - SQL Data Mining, ML & Analytic Functions + R Integration for Scalable, Distributed, Parallel in-DB ML Execution
- Oracle Database 12c

Oracle Cloud

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
## Oracle Advanced Analytics 12.2

**Model Build Time Performance**

### OAA 12.2 Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rows</th>
<th>Model Build Time</th>
<th>T7-4 (Sparc &amp; Solaris)</th>
<th>X5-4 (Intel and Linux)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes Importance</td>
<td>640</td>
<td>28s / 512</td>
<td></td>
<td>44s / 72</td>
</tr>
<tr>
<td>K Means Clustering</td>
<td>640</td>
<td>161s / 256</td>
<td></td>
<td>268s / 144</td>
</tr>
<tr>
<td>Expectation Maximization</td>
<td>159</td>
<td>455s / 512</td>
<td></td>
<td>588s / 144</td>
</tr>
<tr>
<td>Naive Bayes Classification</td>
<td>320</td>
<td>17s / 256</td>
<td></td>
<td>23s / 72</td>
</tr>
<tr>
<td>GLM Classification</td>
<td>640</td>
<td>154s / 512</td>
<td></td>
<td>363s / 144</td>
</tr>
<tr>
<td>GLM Regression</td>
<td>640</td>
<td>55s / 512</td>
<td></td>
<td>93s / 144</td>
</tr>
<tr>
<td>Support Vector Machine (IPM solver)</td>
<td>640</td>
<td>404s / 512</td>
<td></td>
<td>1411s / 144</td>
</tr>
<tr>
<td>Support Vector Machine (SGD solver)</td>
<td>640</td>
<td>84s / 256</td>
<td></td>
<td>188s / 72</td>
</tr>
</tbody>
</table>

The way to read their results is that they compare 2 chips: X5 (Intel and Linux) and T7 (Sparc and Solaris). They are measuring scalability (time in seconds) with increase degree of parallelism (dop). The data also has high cardinality categorical columns which translates in 9K mining attributes (when algorithms require explosion). There are no comparisons to 12.1 and it is fair to say that the 12.1 algorithms could not run on data of this size.
Rapidly Build, Evaluate & Deploy Analytical Methodologies
Leveraging a Variety of Data Sources and Types

**SQL Joins and arbitrary SQL transforms & queries – power of SQL**

**Unstructured data also mined by algorithms**

**Consider:**
- Demographics
- Past purchases
- Recent purchases
- Comments & tweets

**Generates SQL scripts and workflow API for deployment**

**Transactional POS data**

**Modeling Approaches**

**Inline predictive model to augment input data**
Big Data Analytics using w Graph
Oracle Advanced Analytics/Machine Learning with Enhanced Graph & Spatial Data Sources

• Add new engineered features
  – Percentage time spent in zones
  – Amount time/encounters with persons of interest

• Better predictions using available data
  – At risk customers
  – Government approval processes
  – Medical claims
  – IoT predictive analytics

Transactionally network relationships data

Transactional geo-location data summarized to % time spent in areas or number of “hits” near a location

Better data and “engineered features”; better predictive models and predictive insights
More Data Variety—Better Predictive Models

• Increasing sources of relevant data can boost model accuracy

Model with “Big Data” and hundreds -- thousands of input variables including:
• Demographic data
• Purchase POS transactional data
• “Unstructured data”, text & comments
• Spatial location data
• Long term vs. recent historical behavior
• Web visits
• Sensor data
• etc.

engineered features – Derived attributes/variable that reflect domain knowledge—key to best models

Naïve Guess or Random

Model with 20 variables

Model with 75 variables

Model with 250 variables

Population Size

Responders

0%

100%
WHAT ARE THE REQUIREMENTS FOR ENTERPRISE AI?

- Open-source (Linux, Hadoop)
- Scalable, Containerized, Fast
- Integrates With Existing Tech (JVM)
- Cross-Team Solution (DevOps, Data Science)
- General-Purpose, Customizable Framework
SKYMIND GIVES
BIG COMPANIES
DEEP LEARNING
AN OPEN-CORE COMPANY
CLOUDERA FOR AI

- Enterprise Distribution
- Easy Integration with Production Stack
- Supports Major Hardware
- ETL, Training, Inference for DL
USE CASES

- NETWORK INTRUSION DETECTION
- FRAUD/ANOMALY DETECTION
  - PAYMENTS, TELCO, ID
- IMAGE RECOGNITION
- PREDICTIVE ANALYTICS
  - MARKET FORECASTING
HARDWARE ACCELERATION + PRODUCTION JVM STACK

Skymind Intelligence Layer (SKIL)

JAVACPP

SPARK

GPUs cuDNN

CPUs MKL
KEY INTEGRATIONS: SPARK, MESOS, KAFKA & HADOOP
THE SKYMIND INTELLIGENCE LAYER
A COMPLETE AI STACK

Graph Database
(BDSG and Oracle Spatial and Graph)

Data

DataVec
“Rosetta Stone” of vectorization

Hadoop
Spark
Docker

DEEPLEARNING4J
Open-source distributed DL for the JVM

ND4J
Scientific computing for java
(our linear algebra engine)

GPUs
Native

Predictions & Classifications

Swappable & Parallel

Graph Database
(BDSG and Oracle Spatial and Graph)
NETWORK INTRUSION DETECTION (NID) WITH DEEP LEARNING
Network intrusion detection is conceptually simple.
● Data: A sequence of network activity for a machine on a corporate network.
● Goal: To determine if that activity is legitimate or fraudulent.
At a high level, network intrusion is similar to other anomaly detection problems.
- Financial fraud detection
- Breakdown detection in
  o Vehicles (cars, aircraft)
  o Manufacturing equipment
  o Datacenter servers
- Campus security (surveillance video, etc.)
In every case, we have a sequence of activity, most of which is legitimate.
Issues with corporate network security:
- Corporation may have 10s of thousands of machines
- How to monitor them all?
- Breaches are extremely costly
- First line of defence:
  o Network intrusion prevention systems. Firewalls, etc.
- We have to assume those fail, so the challenge is network intrusion detection (NID)
There are 2 basic approaches:

- "Signature based" (we have a labeled dataset of known attacks, supervised learning)
- Anomaly based (we don't know what attacks look like)
UNSUPERVISED NID
(ANOMALY DETECTION)
Pros:

- Doesn’t need labeled data - can build a system based on raw/unlabeled data
- Can detect novel/previously unseen attacks: the "unknown unknowns"
Cons

- More ambiguous: “This is unusual” rather than “p/DDOS\(=0.95\)”
- Watch the false positive rate: "Unusual" doesn’t always mean malicious
Most effective systems make use of both supervised and unsupervised methods, as well as rules engines.
Network Intrusion Detection with Skymind (DL4J) and Big Data Spatial and Graph
• Understand the data
  – UNSW-NB15 data set for **Network Intrusion Detection** systems
    – Created by IXIA PerfectStorm tool in Cyber Range Lab of Australian Centre for Cyber Security
    – A mix of
      – Real modern normal activities, and
      – Synthetic contemporary attack behaviors


Data Cleansing & preparation
Train Neural Network model

- Understand
  - Features of UNSW-NB15 data set

49 original features
• Understand the data. One round of clean up.
  
  – Ports should be all integer based, however, there are **Hex** values
  
  – Action: convert them back to decimal

```
59,166,0,1,62377,149,171,126,4,53,udp,CON,0,001044,130,162,31,29,0,0,dns,498084,2813,620689,625,2,192,168,241,243,259,192,168,241,243,49320,icmp,URH,0,1780,0,64,0,0,0,-.196,4095,0,5,0,0,0,0,0,356,192,168,241,243,49320,192,168,241,243,0xc0a8,icmp,URH,0,1780,0,64,0,0,0,-.196,4095,0,5,0,0,0,0,0,356
59,166,0,6,38993,149,171,126,0,53,udp,CON,0,0,0106,132,164,31,29,0,0,dns,498113,1875,618867,875,2,2
59,166,0,9,59720,149,171,126,8,53,udp,CON,0,00107,132,164,31,29,0,0,dns,493457,9375,613084,125,2,2
59,166,0,4,21489,149,171,126,7,53,udp,CON,0,001144,130,162,31,29,0,0,dns,454545,4688,566433,5625,2
59,166,0,8,45682,149,171,126,0,53,udp,CON,0,001257,130,162,31,29,0,0,dns,413683,375,515513,125,2,2
59,166,0,8,32958,149,171,126,8,53,udp,CON,0,00124,132,164,31,29,0,0,dns,469750,9063,583629,9375,2
59,166,0,8,55879,149,171,126,3,53,udp,CON,0,001075,146,178,31,29,0,0,dns,543255,8125,662325,5625,2
59,166,0,0,43096,149,171,126,3,53,udp,CON,0,001114,132,164,31,29,0,0,dns,473967,6875,588868,9375,2
59,166,0,2,31439,149,171,126,1,53,udp,CON,0,001088,146,178,31,29,0,0,dns,536764,6875,654411,75,2,2
59,166,0,3,45426,149,171,126,0,53,udp,CON,0,001053,132,164,31,29,0,0,dns,501425,5,622981,9375,2,2
59,166,0,9,28993,149,171,126,3,53,udp,CON,0,001173,132,164,31,29,0,0,dns,450127,875,569249,8125,2,
```
• Understand the data & define transformations

```
,removeColumns("timestamp start", "timestamp end", "source ip", "destination ip", "source TCP base sequence num", "dest TCP base sequence num", "attack category")
.filter(new FilterInvalidValues("source port", "destination port")) // Remove example
.transform(new ReplaceEmptyIntegerWithValueTransform("count flow http methods", 0))
.transform(new ReplaceInvalidWithIntegerTransform("count ftp commands", 0)) // Only ftp
.transform(new ConditionalTransform("is ftp login", 1, 0, "service", Arrays.asList("ftp")))
.transform(new ReplaceEmptyIntegerWithValueTransform("count flow http methods", 0))
.transform(new StringToCategoricalTransform("service", ":", "dns", "http", "smtp", "ftp", "udp")
.transform(new MapAllStringsExceptListTransform("transaction protocol", ":", "other", Arrays.asList("FIN", "ACK", "RST", "PUSH", "SYN", "FIN+ACK", "RST+ACK")))
.transform(new StringToCategoricalTransform("state", ":", "FIN", "CON", "INT", "RST", "FIN+ACK", "CON+ACK", "FIN+rst", "CON+rst")
.transform(new IntegerToCategoricalTransform("equal ips and ports", Arrays.asList("not ftp", "ftp login", "ftp data")))
```

Categorical to One Hot transformation

• Service “dns” becomes

```
0 1 0 0 0 0 0 0 0 0 0 0 0 0
```
• Executed transformations with Scala & Apache Spark using Oracle’s Big Data stack

```scala
val stringData = jsc.textFile("/user/oracle/UNSW-complete-all-removedhex.csv");

import org.datavec.spark.transform.AnalyzeSpark;
import org.datavec.spark.transform.SparkTransformExecutor;
import org.datavec.spark.transform.misc.StringToWritablesFunction;

val swf = new StringToWritablesFunction(recordReader);
val parsedInputData = stringData.map(swf);
val processedData = SparkTransformExecutor.execute(parsedInputData, tp);
```

• Save the RDD back to CSV format
• Built a Multi-Layer Perceptron (MLP) Neural Network

```java
conf = new NeuralNetConfiguration.Builder()
    .seed(seed)
    .iterations(iIter)
    .activation(Activation.TANH)
    .weightInit(WeightInit.XAVIER)
    .learningRate(learningRate)
    .regularization(true), 12(1e-4)
    .list()
    .layer(0, new DenseLayer.Builder().nIn(numInputs), nOut(iLayer1), build())
    .layer(1, new DenseLayer.Builder().nIn(iLayer1), nOut(iLayer2), build())
    .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
        .activation(Activation.SOFTMAX)
        .nIn(iLayer2), nOut(outputNum), build())
    .backprop(true), pretrain(false)
    .build();
```
• Tested the quality of MLP NN
  • After 800 iterations of training
    Accuracy: 0.9811
    Precision: 0.9894
    Recall: 0.9286
    F1 Score: 0.958
    • Labeled as “non-intrusion” classified as “non-intrusion”: 46 times
    • Labeled as “intrusion” classified as “non-intrusion”: 1 time
    • Labeled as “intrusion” classified as “ intrusion”: 6 times
      \[
      \frac{(46+6)}{46+6+1} = 0.9811
      \]
  • Long Short-Term Memory (LSTM) NN gave similar F1 result
• A Single GPU: GTX 970 (1664 CUDA cores, 4GB device RAM)
• 2-Quad core Intel CPUs (Xeon E5620 2.4GHz)
• CUDA 7.5
• Converted CSV to a Property Graph (Oracle defined flat file .opv/.ope)
  • Model each IP as a vertex
  • Model each record (traffic from a source IP to a destination IP) as an edge
  • 60+ Features become properties of edges
• Utility provided in BDSG
  • OraclePropertyGraphUtilsBase.convertCSV2OPV
  • OraclePropertyGraphUtilsBase.convertCSV2OPE

Example CSV file
1,John,4.2,30
2,Mary,4.3,32
3,"Skywalker, Anakin",5.0,46
4,"Darth Vader",5.0,46
5,"Skywalker, Luke",5.0,53

Example output .opv file
1,name,1,John,,
1,score,4,,4.2,,
1,age,2,,30,,
2,name,1,Mary,,
2,score,4,,4.3,,
2,age,2,,32,
• Utilized the built-in *parallel* graph data loader

• A single API call to `loadData` method

```java
OraclePropertyGraphDataLoader opgdl =
OraclePropertyGraphDataLoader.getInstance();

opgdl.loadData(opg,

"<PATH>/net_intrusion.opv",
"<PATH>/net_intrusion.ope",
8 // 8 threads
);
```
• Network Intrusion Detection
  Property Graph
  • Blue edges: malicious
  • Other edges: normal traffic
  • Many attacks originated from
    175.45.176.1 to target
    149.171.126.17
  • Visualization tool: Cytoscape v3.2.1
    + Big Data Spatial and Graph v2.1
• Focused on “Attacks” graph
• Focused on “Attacks” graph
• Focused on “Attacks” graph

• Applied built-in analytics in BDSG

• Found top-3 IP addresses with **highest Page Rank** value
Q & A
Resources

• Oracle Spatial and Graph
  oracle.com/technetwork/database/options/spatialandgraph

• Oracle Big Data Spatial and Graph

• Skymind
  http://skymind.io  http://deeplearning4j.org

• Oracle Advanced Analytics