Real-time Fraud Detection with Innovative Big Graph Feature

Gaurav Deshpande, VP Marketing, TigerGraph; Mingxi Wu, VP Engineering, TigerGraph
Speaking Today

Gaurav Deshpande
VP Marketing, TigerGraph
gaurav@tigergraph.com
B. Tech. IIT Bombay, M. S. OSU
Big Data Analytics Veteran, 14 patents in supply chain management and analytics

Mingxi Wu
VP Engineering, TigerGraph
Mingxi@tigergraph.com
Ph.D. in Database & Data Mining, Univ. of Florida
Turn, Oracle, Microsoft, patents on big data & graph management
Agenda

1. Fraud and Money Laundering – Scale and Complexity

2. Real-time anti-scam system architecture
   - 460 million phones, 10B+ call graph, 1000s new calls per second

3. Innovative graph features for effectively combating phone scam
   - 118 new features per phone – new training data for machine learning fed every 2 hours

4. GSQL - TigerGraph's innovative SQL-like query language
Fraud Impacts Multiple Industries

63% of businesses have experienced the same or more fraud losses in the past 12 months

Online fraud alone costs consumers $16 Billion per year (bank and merchant costs are higher)

$300 Billion in global loss across Telecom from uncollected revenue and fraud in 2016
Consider an Example - Phone Scam

Illegally acquiring money from victims, or failing to pay a telecom company

- $4.96 Billion – Compromised PBX/Voicemail Systems
- $4.32 Billion – Subscription/Identity Theft
- $3.84 Billion – International Revenue Share Fraud
- $2.88 Billion – By-Pass Fraud
- $2.40 Billion – Credit Card Fraud
Current Approach

- **Data:** features of entities, e.g., users, accounts, locations. Examples:
  - Phone-based fraud detection: frequency and duration of one-directional calls
- **Detection:** Analysts manually write rules regarding features of nodes or their immediate neighbors (1 to 2 hops).
- **Performance**
  - False positives: too many cases to investigate, block legitimate business
  - False negatives: fail to catch many fraud cases
- **Using Machine Learning for Fraud Detection - Imbalanced Dataset Classification**
  - Less than 1% of total call volume is related to confirmed phone scam activity
Combat Fraud
The Challenges

- **Complex modeling**
  - Customer 360, with ever growing schemas

- **Real-time monitoring**
  - Time sensitive
  - Post-event detection is useless

- **Large-scale event/transaction handling**
  - Billions or trillions scale

- **Deep link analysis**
  - Go beyond immediate neighbors to discover non-obvious relationships

- **Complex contextual analysis**
  - Subtle and effective signal buried in network

- **Complex logic**
  - Complicated, expensive coding with existing SQL or graph languages
Detecting Phone-Based Fraud by Analyzing Network or Graph Relationship Features

**Good Phone Features**
1. High call back phone
2. Stable group
3. Long term phone
4. Many in-group connections
5. 3-step friend relation

**Bad Phone Features**
1. Short term call duration
2. Empty stable group
3. No call back phone
4. Many rejected calls
5. Average distance > 3
6. Empty stable group
7. Many rejected calls

**Diagram:**
- **Good Phone:**
  - Phone 1
  - Phone 2
  - Phone 3
  - Phone 4
  - Phone 5
  - Phone 6
  - Phone 7
  - Phone 8
  - Phone 9
  - Phone 10
  - Phone 11
  - Stable group
  - Many in-group connections

- **Bad Phone:**
  - Phone 1
  - Phone 2
  - Phone 3
  - Phone 4
  - Phone 5
  - Phone 6
  - Phone 7
  - Phone 8
  - Phone 9
  - Phone 10
  - Empty stable group
  - Many rejected calls
  - Average distance > 3

© 2018 TigerGraph. All Rights Reserved
### Improving accuracy for machine learning

<table>
<thead>
<tr>
<th></th>
<th>Prankster</th>
<th>Regular Customer</th>
<th>Fraudster</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tim</strong></td>
<td>[Image]</td>
<td></td>
<td>[Image]</td>
<td></td>
</tr>
<tr>
<td><strong>Sarah</strong></td>
<td>[Image]</td>
<td></td>
<td>[Image]</td>
<td></td>
</tr>
<tr>
<td><strong>Fred</strong></td>
<td>[Image]</td>
<td></td>
<td>[Image]</td>
<td></td>
</tr>
<tr>
<td><strong>John</strong></td>
<td>[Image]</td>
<td></td>
<td>[Image]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>Tim</th>
<th>Sarah</th>
<th>Fred</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age of sim card</td>
<td>2 weeks</td>
<td>4 weeks</td>
<td>3 weeks</td>
<td>2 weeks</td>
</tr>
<tr>
<td>% of one directional calls</td>
<td>50%</td>
<td>10%</td>
<td>55%</td>
<td>60%</td>
</tr>
<tr>
<td>% rejected calls</td>
<td>30%</td>
<td>5%</td>
<td>28%</td>
<td>25%</td>
</tr>
<tr>
<td><strong>Prediction by ML with call history features</strong></td>
<td>Likely Fraudster</td>
<td>Regular Customer</td>
<td>Likely Fraudster</td>
<td>Likely Fraudster</td>
</tr>
<tr>
<td>Stable group</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Many in-group connections</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>3-step friend relation</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Prediction by ML with deep link Graph features</strong></td>
<td>Likely Prankster</td>
<td>Regular Customer</td>
<td>Likely Fraudster</td>
<td>Likely Sales</td>
</tr>
</tbody>
</table>

© 2018 TigerGraph. All Rights Reserved
Generating New Training Data for Machine Learning to Detect Phone-Based Scam

Graph with 460 Million phones and 10 Billion calls, 1000s of new calls per second. Feed Machine Learning with new training data with 118 features per phone every 2 hours.

Phone 1 Features
(1) High call back phone
(2) Stable group
(3) Long term phone
(4) Many in-group connections
(5) 3-step friend relation

Phone 2 Features
(1) Short term call duration
(2) Empty stable group
(3) No call back phone
(4) Many rejected calls
(5) Avg. distance > 3

Machine Learning Solution

Tens – Hundreds of Billions of calls
Combat Phone Scam In Real-Time

**CHINA MOBILE** online telecommunication anti-scam system

**Facts**
- **460M** vertices (inside & outside network phone numbers)
- **10B+** edges (each edge stores aggregate info of calls between two phones)
- **300M** daily real-time post
- **2K** calls per second per machine (2k is the peak QPS, in real life it seldom hit it, $2000 \times 3600 \times 24 \times 5 = 864M/day$ capacity)
- [Single server 32 COREs, 64 threads + 512G Memory] x 5

**Single Machine Provides**
- **Sub-second** graph feature collection.
- **2 hours** batch collecting **2M** calls $\times 118$ features training data
- Real-time update to call graph
Graph Features - Simple Neighbor Aggregate

AvgAccum<float> @@avgDuration;
MaxAccum<float> @@maxDuration;
MinAccum<float> @@minDuration;
SumAccum<float> @@totalDuration;
SumAccum<float> @@stdDeviation;
SumAccum<int> @@totalCallCount;

Seed = {phoneNumber};

TargetGroup = SELECT tgt
FROM   Seed:src-(:e)-phone:tgt
WHERE  tgt.outdegree() < 1000 AND e.date BETWEEN "2018/01/01" AND "2018/01/31"
ACCUM  @@avgDuration += e.duration,
       @@maxDuration += e.duration,
       @@minDuration += e.duration,
       @@totalDuration += e.duration,
       @@totalCallCount += 1;

TargetGroup = SELECT tgt
FROM   Seed:src-(:e)-phone:tgt
WHERE  tgt.outdegree() < 1000 AND e.date BETWEEN "2018/01/01" AND "2018/01/31"
ACCUM  @@stdDeviation += (e.duration - @@avgDuration)^2;

@@stdDeviation = SQRT(@@stdDeviation/@@totalCallCount);
Addressing Fraud Prevention and AML compliance with Real-time Deep Link Analytics

Real-time Multi-Hop Performance
Sub-second response for queries touching tens of millions of entities/relationships

Transactional (Mutable) Graph
Hundreds of thousands of updates per second, Billions of transactions per day

Scalability for Massive Datasets
100 B+ entities, 1 Trillion+ relationships

Deep Link Analytics
Queries traverse 3 to 10+ hops deep into the graph performing complex calculations

Privacy for Sensitive Data
Control access based on user role, data type, or department

Ease of Development & Deployment
Easy to use query language (GSQL) for rapidly developing & deploying complex analytics
Real-Time Graph Analytics Platform

User Interface:
- GSQL language for schema/loading/queries/updates
- REST API for connecting to other applications
- GraphStudio GUI for human interaction

Output:
- JSON or visual graph
Combat Fraud
Solution Highlights

- **Complex modeling**
  - Graph model
- **Real-time monitoring**
  - real-time post new data
  - real-time query response
- **Large-scale event/transaction handling**
  - scale out and up
  - high compression

- **Deep link analysis**
  - Analyze 3-10+ hops across the graph
- **Complex contextual analysis**
  - GSQL, customer delighted comparing with Cypher & Gremlin
- **Complex logic**
  - Complicated, expensive coding with existing SQL or graph languages
Detecting Fraud with TigerGraph

Real-time Deep Link Analytics at Massive Scale

• **Graph Features to the rescue**
  • Integrate multiple data sources into one graph
  • Real-time updates
  • GSQL helps easily collect complex, deep-link, aggregate graph features
  • Feed Machine Learning algorithm with new training data

• **Deep Link Analytics to the rescue**
  • GSQL easily describes graph traversal and compute patterns
  • Massive parallel processing for speed and efficiency
  • Visualization shows evidence right on the spot
## Customers

<table>
<thead>
<tr>
<th>Company</th>
<th>Description</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alipay</td>
<td>#1 e-payment company in the world, 100M daily active users</td>
<td>The largest transaction graph in production in the world (100B+ vertices, 2B+ daily real time updates)</td>
</tr>
<tr>
<td>VISA</td>
<td>#1 US payment company</td>
<td>Business Graph</td>
</tr>
<tr>
<td>wish</td>
<td>#1 Mobile E-commerce</td>
<td>Real-time Personalized Recommendation</td>
</tr>
<tr>
<td>UBER</td>
<td>#1 Ride sharing company</td>
<td>Risk and Compliance</td>
</tr>
<tr>
<td>elementum</td>
<td>#1 Mobile global supply-chain</td>
<td>Supply-chain logistics</td>
</tr>
<tr>
<td>State Grid</td>
<td>#1 Power-grid company</td>
<td>Electrical Power Grid</td>
</tr>
</tbody>
</table>
### Benchmark

#### OLTP queries (6.5 B vertices, 32B edges)

<table>
<thead>
<tr>
<th>Maximum Traversal Steps</th>
<th>Query Name</th>
<th>QPS (Queries per Second)</th>
<th>Latency/Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVG (ms) MIN (ms)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>3606 55 8</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>2</td>
<td>3831 44 11</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>4928 40 10</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>127 1199 18</td>
</tr>
</tbody>
</table>

#### OLTP queries (20B vertices, 95B edges)

<table>
<thead>
<tr>
<th>Maximum Traversal Steps</th>
<th>Query Name</th>
<th>QPS (Queries per Second)</th>
<th>Latency/Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVG (ms) MIN (ms)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>2096 107 21</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>2</td>
<td>2286 129 32</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>2789 84 26</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>247 1181 39</td>
</tr>
</tbody>
</table>

#### OLAP query (20B vertices, 95B edges)

<table>
<thead>
<tr>
<th>Query Name</th>
<th>Iterations</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>10</td>
<td>36</td>
</tr>
</tbody>
</table>
Backup
Evolution of Graph Technology

**Graph 1.0**
Example - Neo4j
- Storage and Visualization focused
- No built-in parallel computation model
- Very slow loading large datasets
- Cannot scale out
- Not designed for real-time graph updates or queries for large datasets
- Limited multi-hop analytics capabilities on large graphs (2 hops)

**Graph 2.0**
Example - DataStax
- Better scale-out, but speed and updates are still an issue
- Built on top of NoSQL repository such as Apache Cassandra
- Not designed for real-time graph updates or queries for large datasets
- Limited multi-hop analytics capabilities on large graphs (2 hops)

**Graph 3.0**
Example - TigerGraph
- Scalability for massive datasets
- Supports real-time graph updates and queries for enterprise scale
- Provides deep link analytics (3-10+ hops) traversing millions of nodes and performing complex calculations
- Privacy for sensitive data
- Ease of use for development & deployment