TigerGraph

Graph Databases and Machine Learning: Finding a Happy Marriage

Victor Lee, November 12, 2018

Know Your Speaker

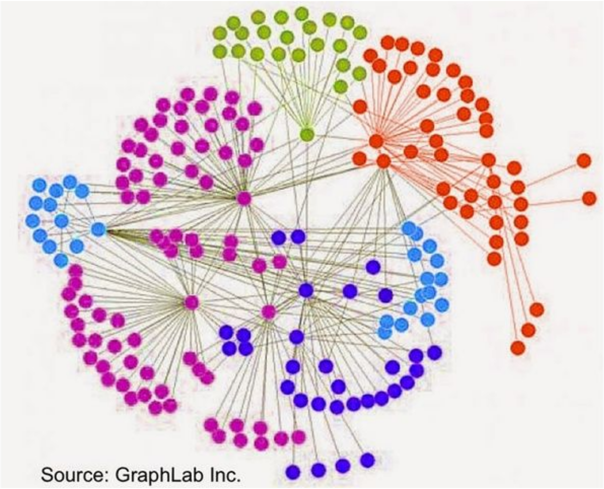
Victor Lee, Director of Product Management
PhD in Graph Data Mining
MS Electrical Engineering, Stanford
BS EE/CS, UC Berkeley

TigerGraph

- **Mission**: Unleash the power of interconnected data for deeper insights and better outcomes
- **Technology**: Industry's First and Only Native Massively Parallel Processing(MPP) Graph Technology
- **Product**: The world's fastest graph database used by organizations including AliPay, Intuit, Uber, Visa, Zillow

Two Hot Items: A Good Match?

Graph Database



New, exciting way to represent information and to query it.

Machine Learning

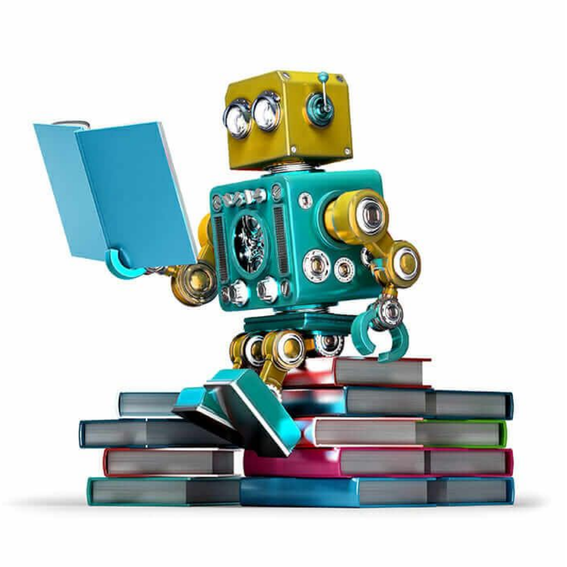
Established powerhouse for predictions and "smart" systems, but still tricky to use.

*Graph analysis is possibly the **single most effective competitive differentiator** for organizations pursuing data-driven operations and decisions after the design of data capture."*

Gartner

What is Machine Learning?

- Branch of AI
- Making predictions, models, or optimizations using incomplete or inexact data
  - Model? Weather
  - Optimization?
    Best driving route,
    best investment portfolio



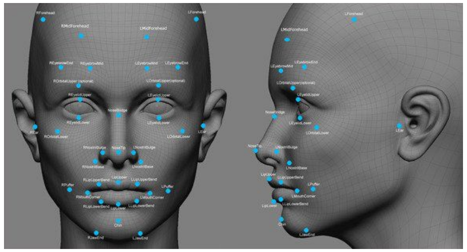
TigerGraph

Uses of Machine Learning

- Virtual Personal Assistants
  - Voice-to-text
  - Semantic analysis
  - Formulate a response

- Real-time route optimization
  - Waze, Google Maps

- Image Recognition
  - Facial recognition
  - X-ray / CT scan reading



- Effective spam filters

TigerGraph

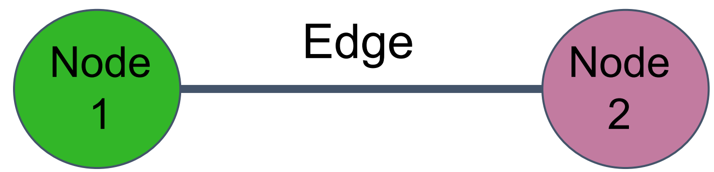
Machine Learning Techniques

- **Supervised Learning**
  Humans provide "training data" with known correct answers
  - Decision Trees
  - Nearest Neighbor
  - Hidden Markov Model,Naïve Bayesian
  - Linear Regression
  - Support Vector Machines (SVM)
  - Neural Networks

- **Unsupervised Learning**
  No "correct" answer; detect patterns or summarize
  - Clustering, Partitioning, Outlier detection
  - Neural Networks
  - Dimensionality reduction

Graphs (Networks) are all around us

Graph: collection of **nodes** and links ("**edges**") between nodes

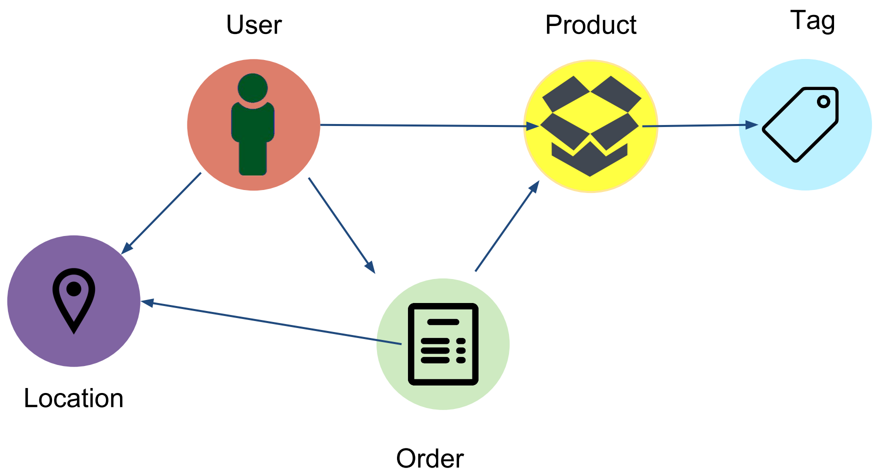


- Data Never Sleeps
- Internet of Things
- BIG Data
- Social Networks

TigerGraph

Graph Database - The Natural Choice for Interconnected Data

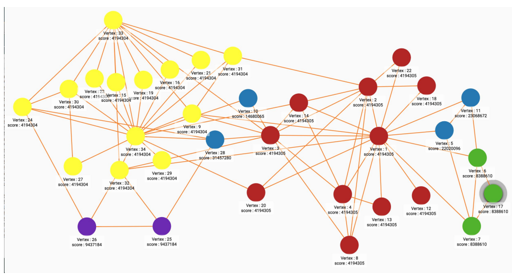
- **Natural storage model** for interconnected data
- **Natural for transactions**: transaction = edge
- **Natural for computational knowledge/inference/ learning** – "connect the dots"



Matchmaking: Consider Each Party

Graph Databases

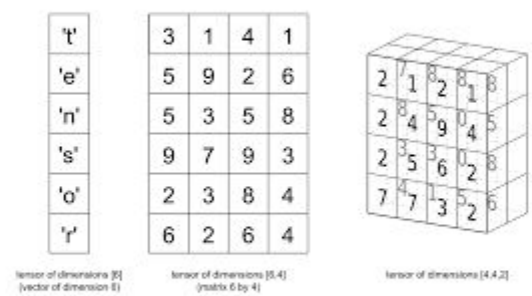
- Stores data as nodes & edges.
- An edge (link) IS a data object.



- Capabilities of platforms and query languages vary:
    - All are good a storing data
    - Wide range of analytical abilities

Machine Learning (ML)

- Wants data as vector, array, or tensor



- Computationally intensive, long and time-consuming
- Requires good quality input data:
  Garbage in, garbage out
- Many methods to choose from

ML Features and Modeling

- Most/All ML methods try to correlate *features* (properties/attributes) with the target result.

	Feat 1	Feat 2	Feat 3	**Result**
Item 1	X	0	0	**X**
Item 2	X	X	0	**0**
Item 3	X	0	X	**?**

- Quality of modeling depends on quality of features
  - Having the right features
  - Having the right distribution of values & results

- Don't know which features are needed!

ML Features and Modeling

- If you don't know the right features yet, then just try to have LOTS of features.

- Big Data 3 V's:
  - **Volume**
  - **Variety**
  - Velocity

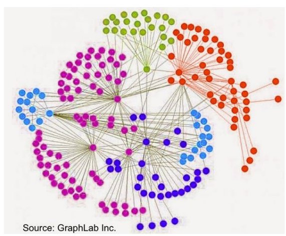
- Example:
  - Suppose family medical history is the best predictor of an ailment.
  - If you don't know family medical history, you won't be able to make good predictions

3 Possible Arrangements for Graph DB + ML

1. Graph Data leaves home and moves to ML.
2. ML moves into the Graph Database.
3. A Graph Database and ML partnership.

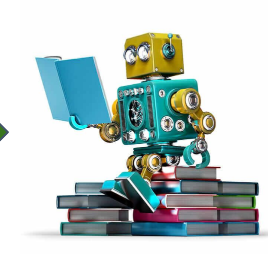
#1 - Graph Data leaves home and moves to ML

- Export relevant graph data
- Simple analogy to Relational Database:
  - Table(s) of node data
  - Table(s) of edge data



NodeID	attr1	attr2	attr3

EID	NID1	NID2	Eattr1	Eattr2	Eattr3



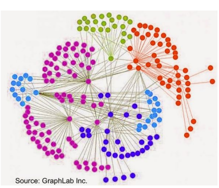
- Traditional approach
  - ML is fed data from DBs or just data files

- Business Value
  - Easy to deploy
  - Graph DB still valuable for non-ML queries and analytics

Exporting Graph Data to ML: Concerns

- Is exported data really "flat" enough?
  - Edge records refers to Node records
  - ML algorithms usually want records to be independent
  - One answer is **graph embedding**

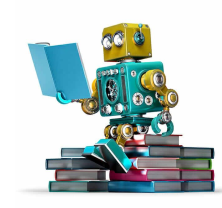
- Graph Embedding
  "Mapping a graph's nodes and edges to points in space."
  - Example: drawing a graph on paper is mapping to 2D space.
  - General graphs can be drawn in 3D space → tensor



NodeID	attr1	attr2	attr3

EID	NID1	NID2	Eattr1	Eattr2	Eattr3

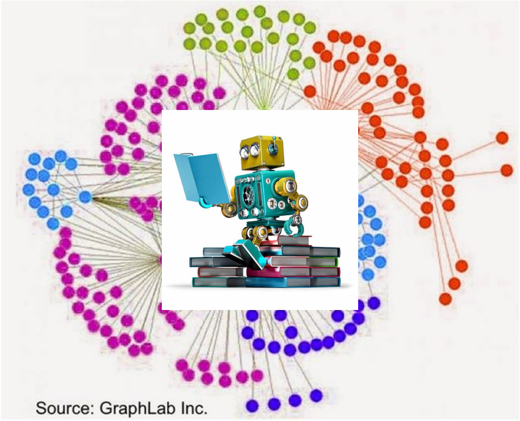
Graph Embedding



TigerGraph

#2 - ML moves into the Graph Database

- Implement ML algorithm as an advanced query
- Novel Approach:
  in-graph analytics



- Business Value
  - Integration
  - Eliminate need for separate systems

ML in Graph Database: Concerns

1. Computational power of Graph DB
2. Graph query language's algorithmic expressiveness
3. Expressing ML algorithms in Graph terms

Only 3rd Gen. Native Parallel Graphs satisfy 1 and 2.

TigerGraph

Evolution of Graph Databases

	Graph 1.0 single server, non-parallel	Graph 2.0 NoSQL base for storage	Graph 3.0 Native, Parallel
Native Graph Storage	✓	✗	✓
Parallel Loading	✗ Days to load terabytes	✗ Days to load terabytes	✓ Hours to load terabytes
Parallel Multi-Hop Analytics	✗ Times out after 2 hops	✗ Runs out of time/memory after 2 hops	✓ Sub-second across 10+ hops
Parallel Updates (in real-time)	✗	✗	✓ Mutable/ Transactional
Scale up to Support Query Volume	✗	✓ Scales for simple queries (1-2 hops)	✓ 2 billion+/day in production
Privacy for Sensitive Data	✗	✗	✓ MultiGraph service

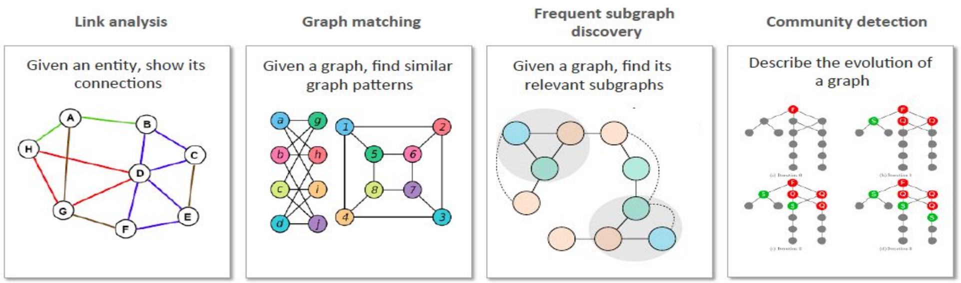
Graph Query Languages

- **SparQL (RDF):** Designed for semantic reasoning, but not computationally intensive work
- **Gremlin (Apache)**: Older scheme. Declarative. Very awkward to write complex tasks.
- **Cypher (Neo4j):** Known by many. Okay for analytics if you couple it with Java.
- **GSQL (TigerGraph)**: Designed for big data analytics.
  - Built-in parallelism and accumulation.
  - Syntax is natural and familiar for algorithms.

TigerGraph

Example: Graph Algorithm Libraries

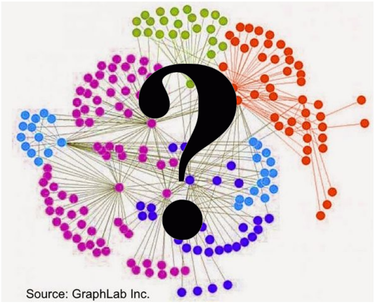
- Some Graph Databases provide libraries of standard graph algorithms
  - Measure/discover characteristics of a graph
  - PageRank, Community Detection, Shortest Path, etc.

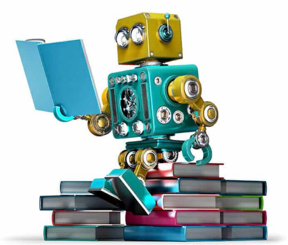


- **GSQL** is well-suited for algorithms
  - Library is written in GSQL
  - Users can see the code and modify as desired
- **Cypher** is less well-suited
  - Library is pre-compiled function calls
  - User can't see how the algorithms are written or modify them

Expressing ML Algorithms in Graph Terms

- Not widely known how to execute most ML algorithms on a graph.





- Same problem as with exporting the Graph data:
  - ML is used to flat data
  - How to make edges a benefit instead of a hinderance?

- This marriage proposal needs further work.

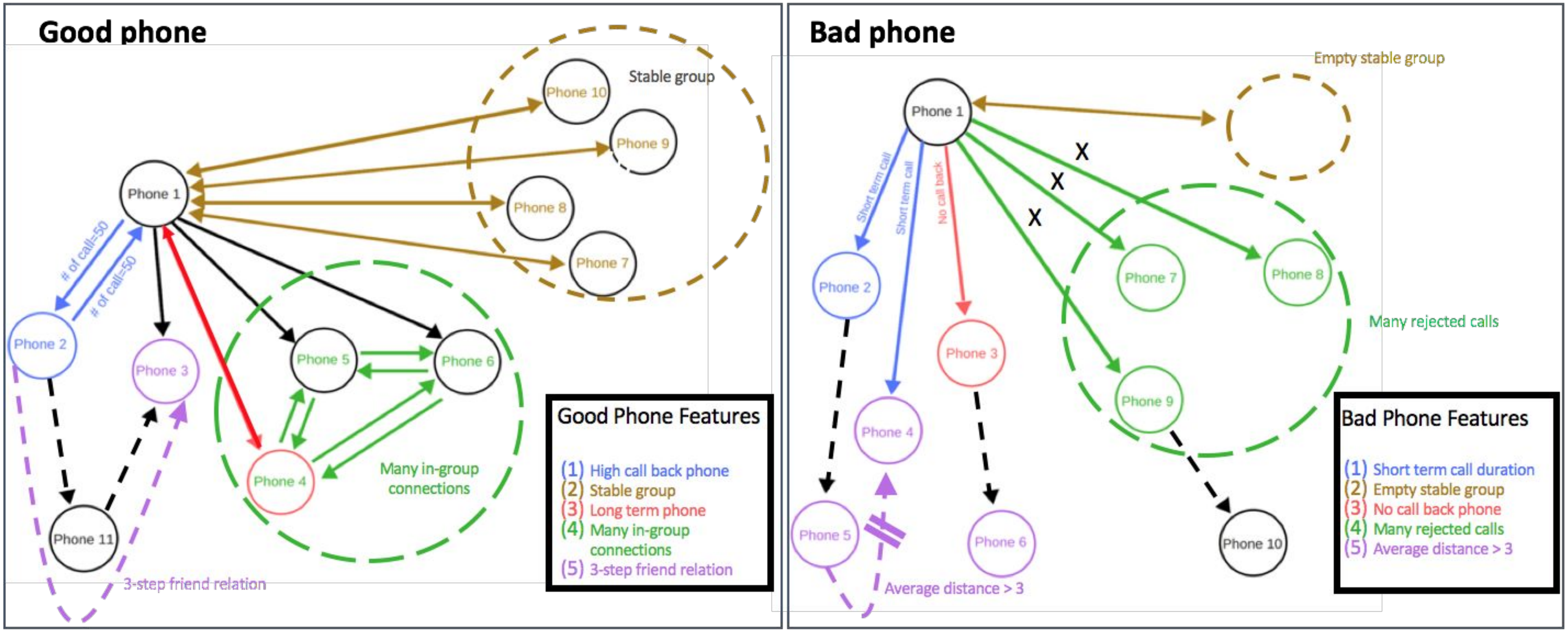
#3 - A Graph Database and ML partnership

- Multi-Step process:
    a. First, Graph DB runs queries to extract graph-based features
    b. Send graph features to ML system.
    c. ML system, enriched by new graph features, learns a predictive model.
    d. Model can be applied back to graph for real-time prediction.

- Seems more like a transactional partnership than a marriage.

TigerGraph

Example: China Mobile Fraud Detection using Graph Features for ML

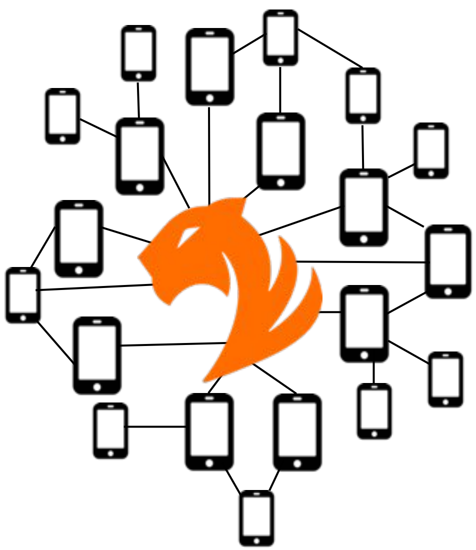


[Download the solution brief at - https://info.tigergraph.com/MachineLearning](https://info.tigergraph.com/MachineLearning)

1

Generating New Training Data for ML to Detect Phone-Based Scam

Graph with 600M phones and 15B call edges,1000s of calls/second.
Feed ML with new training data with 118 features per phone.



Tens – Hundreds of Billions of calls

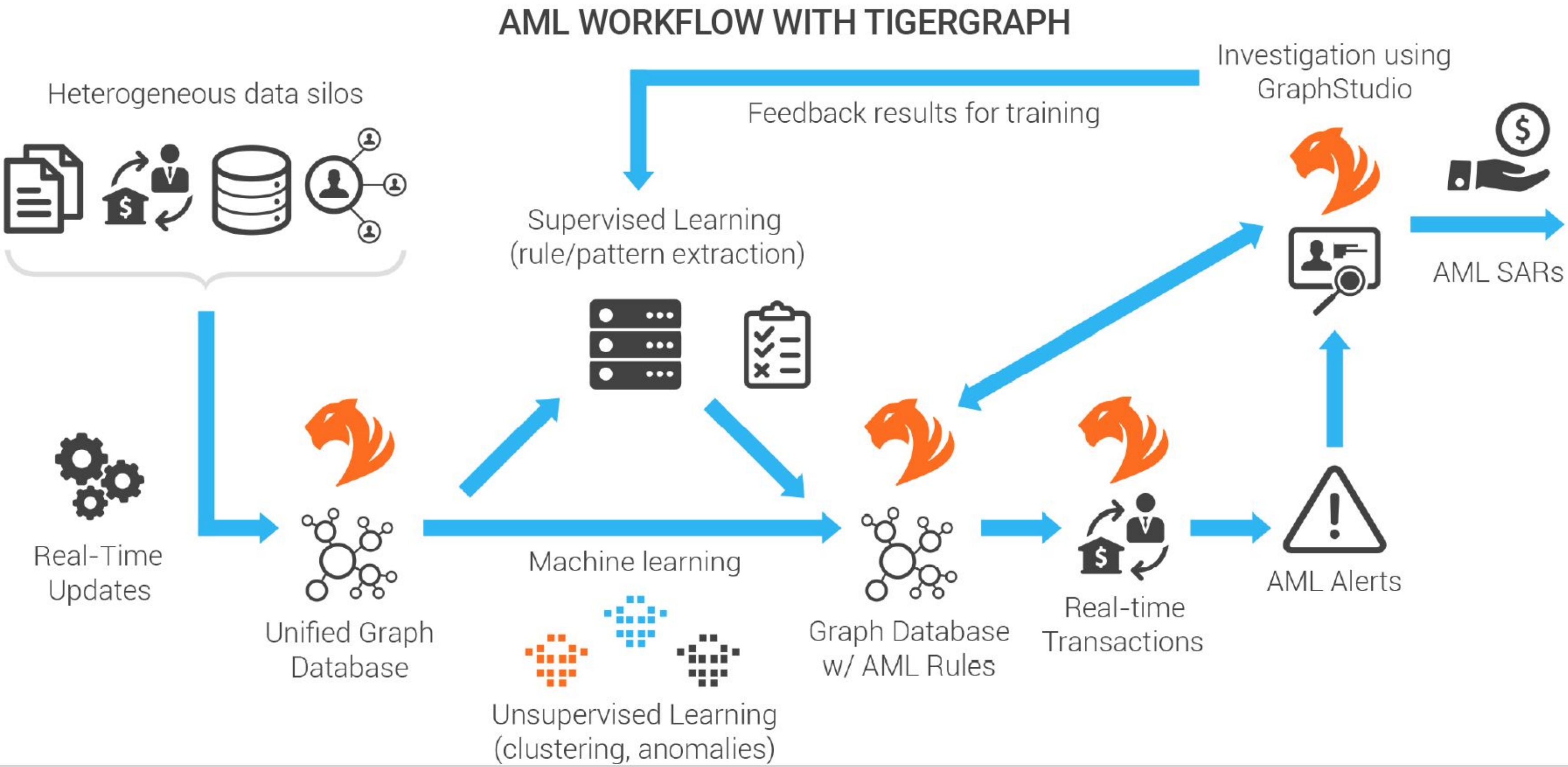
Phone 1 Features
(1) High call back phone
(2) Stable group
(3) Long term phone
(4) Many in-group connections
(5) 3-step friend relation

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

Training Data
118 per phone x 600 Million phones = 70 Billion new features

Machine Learning Solution

Graph-enhanced workflow for Anti-Money Laundering



Summary

Graph Databases and Machine Learning both represent powerful tools for getting more value from data.

3 Proposals for Marrying Graph DBs + ML:

- Export Graph Data to ML system
  - Conventional, but not clear how to treat the data.
- Perform ML within Graph Database
  - Need fast, scalable analytical Graph DB. ML methods not yet ported to Graph DB.
- Export Graph Features to ML system
  - Improves ML results; in practice now.

TigerGraph

Final Thoughts

- Room for diversity.
- Healthy marriages evolve over time.

THANK YOU

TigerGraph