Big Data and Growth in Graph Databases
Graph Connect September 2018

(graphs) – [:ARE] -> (everywhere)
76% Fortune 100 have piloted or adopted Neo4j Graph Database
  • Finance – 20 of top 25
  • Insurance – 8 of top 10
  • Software – 7 of top 10
  • Logistics – 3 of top 5
  • Retail – 7 of top 10
  • Airlines – 3 of top 5
  • Telco – 4 of top 5
  • Hospitality – 3 of top 5
  • Health – spread of disease; cancer research
Graph Database Top Use Cases
Graph Connect September 2018

• Real Time Recommendations
• Fraud Detection
• Network and IT Operations
• Master Data Management – Customer 360
• Knowledge Graph
• Identity and Access Management
“We send email to people so they will visit our web-site and buy our product”

MATCH (e:Email) - [SENT_TO] -> (p:Person {fullName: ‘Steve Newman’}) - [:VISITED] -> (w:Website) <- [:SOLD_ON] - (pr:Product) <-[:Purchased]-(p) RETURN *
Origins of Mathematics of Graph Theory

“Solutio problematis ad geometriam situs pertinentis” – Leonard Euler 1736

Walking the Bridges of Königsberg
4 Main areas of Königsberg with 7 Bridges.
Can you cross each bridge only once and return to your starting point?

Euler’s Insight
The only relevant data is the main areas and the bridges connecting them.

Origins of Graph Theory
Euler abstracted the problem and created generalized rules based on nodes and relationships that apply to any connected system.
“Systemic delay propagation in the US airport network” - Fleurquin, Ramasco & Eguiluz, 2013
Network Science and Graph Algorithms

Most graph queries consider specific parts of the graph (e.g., a starting node), and the work is usually focused in the surrounding subgraph. We term this type of work graph local. This type of graph-local processing is often utilized for real-time transactions and pattern-based queries.

When speaking about graph algorithms, we are typically looking for global patterns and structures. The input to the algorithm is usually the whole graph, and the output can be an enriched graph or some aggregate value such as a score. This approach sheds light on the overall nature of a network through its connections. Organizations tend to use graph algorithms to model systems and predict behaviour based on how things disseminate, important components, group identification, and the overall robustness of the system.
Network Science, statistics, real-world networks
Non-randomness, Power Law, Preferential Attachment

![Graph showing Power-Law Distribution](Image)

- **Power-Law Distribution**
  - Most nodes have very few relationships but a small number of nodes have a lot, which creates hub-and-spoke structures.

- **Power-Law Distribution**
  - Most nodes have the same number of relationships, which creates flat structures.

This is the underinvested area where most nodes really exist. Many statistical models erroneously focus here, on averages.
Network Science and graph analytics
Types of questions graph analytics answer

- Propagation Pathways: How do things spread?
- Flow & Influence: What are the capacities, costs, and control points?
- Interactions & Resiliency: How do things interact and will that change?
Network Science and graph analytics
Types of questions graph analytics answer

- Investigate the route of a disease or a cascading transport failure.
- Uncover the most vulnerable, or damaging, components in a network attack.
- Identify the least costly or fastest way to route information or resources.
- Predict missing links in your data.
- Locate direct and indirect influence in a complex system.
- Discover unseen hierarchies and dependencies.
- Forecast whether groups will merge or break apart.
- Find bottlenecks or who has the power to deny/provide more resources.
- Reveal communities based on behaviour for personalized recommendations.
- Reduce false positives in fraud and anomaly detection.
- Extract more predictive features for machine learning.
Integrating OLTP and OLAP Processing
HTAP (Hybrid Transaction Analytical Processing)

“Ultimately, HTAP will become a key enabling architecture for intelligent business operations.” – Gartner 2018

Graph Algorithms by Amy E. Hodler and Mark Needham (O’Reilly). Copyright 2019 Amy E. Hodler and Mark Needham, 978-1-492-05781-9
Neo4j Graph Database Platform Announced
Graph Connect September 2017

Graph Algorithms by Amy E. Hodler and Mark Needham (O'Reilly). Copyright 2019 Amy E. Hodler and Mark Needham, 978-1-492-05781-9
Databricks Apache Spark Analytics Engine for Big Data Processing includes GraphFrames 2016
The Spark Graph Project is a joint initiative from Apache project contributors in Databricks and Neo4j to bring support for DataFrames, Cypher, and DataFrames based algorithms into the core Apache Spark project as part of the 3.0 release.
<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>What it does</th>
<th>Example use</th>
<th>Spark example</th>
<th>Neo4j example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth First Search</td>
<td>Traverses a tree structure by fanning out to explore the nearest neighbors and then their sublevel neighbors</td>
<td>Locating neighbor nodes in GPS systems to identify nearby places of interest</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Depth First Search</td>
<td>Traverses a tree structure by exploring as far as possible down each branch before backtracking</td>
<td>Discovering an optimal solution path in gaming simulations with hierarchical choices</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Shortest Path Variations: A*, Yen’s</td>
<td>Calculates the shortest path between a pair of nodes</td>
<td>Finding driving directions between two locations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>All Pairs Shortest Path</td>
<td>Calculates the shortest path between all pairs of nodes in the graph</td>
<td>Evaluating alternate routes around a traffic jam</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Single Source Shortest Path</td>
<td>Calculates the shortest path between a single root node and all other nodes</td>
<td>Least cost routing of phone calls</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Minimum Spanning Tree</td>
<td>Calculates the path in a connected tree structure with the smallest cost for visiting all nodes</td>
<td>Optimizing connected routing, such as laying cable or garbage collection</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Random Walk</td>
<td>Returns a list of nodes along a path of specified size by randomly choosing relationships to traverse.</td>
<td>Augmenting training for machine learning or data for graph algorithms.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
# Graph Algorithms – Centrality

## Neo4j and Spark Graph Frames - April 2019

<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>What it does</th>
<th>Example use</th>
<th>Spark example</th>
<th>Neo4j example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Centrality</td>
<td>Measures the number of relationships a node has</td>
<td>Estimating a person’s popularity by looking at their in-degree and using their out-degree to estimate gregariousness</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Closeness Centrality</td>
<td>Calculates which nodes have the shortest paths to all other nodes</td>
<td>Finding the optimal location of new public services for maximum accessibility</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Variations: Wasserman and Faust, Harmonic Centrality</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Betweenness Centrality</td>
<td>Measures the number of shortest paths that pass through a node</td>
<td>Improving drug targeting by finding the control genes for specific diseases</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Variation: Randomized-Approximate Brandes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PageRank</td>
<td>Estimates a current node’s importance from its linked neighbors and their neighbors (popularized by Google)</td>
<td>Finding the most influential features for extraction in machine learning and ranking text for entity relevance in natural language processing.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Graph Algorithms – Community Detection

### Neo4j and Spark Graph Frames - April 2019

<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>What it does</th>
<th>Example use</th>
<th>Spark example</th>
<th>Neo4j example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle Count</td>
<td>Measures how many nodes form triangles and the degree to which nodes tend to cluster together</td>
<td>Estimating group stability and whether the network might exhibit “small-world” behaviors seen in graphs with tightly knit clusters</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Clustering Coefficient</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Connected</td>
<td>Finds groups where each node is reachable from every other node in that same group following the direction of relationships</td>
<td>Making product recommendations based on group affiliation or similar items</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Components</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connected Components</td>
<td>Finds groups where each node is reachable from every other node in that same group, regardless of the direction of relationships</td>
<td>Performing fast grouping for other algorithms and identify islands</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Label Propagation</td>
<td>Infers clusters by spreading labels based on neighborhood majorities</td>
<td>Understanding consensus in social communities or finding dangerous combinations of possible co-prescribed drugs</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Louvain Modularity</td>
<td>Maximizes the presumed accuracy of groupings by comparing relationship weights and densities to a defined estimate or average</td>
<td>In fraud analysis, evaluating whether a group has just a few discrete bad behaviors or is acting as a fraud ring</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
“Artificial intelligence (AI) has undergone a renaissance recently, making major progress in key domains such as vision, language, control, and decision-making. This has been due, in part, to cheap data and cheap compute resources, which have fit the natural strengths of deep learning.

However, many defining characteristics of human intelligence, which developed under much different pressures, remain out of reach for current approaches. In particular, generalizing beyond one’s experiences—a hallmark of human intelligence from infancy—remains a formidable challenge for modern AI...

We discuss how graph networks can support relational reasoning and combinatorial generalization, laying the foundation for more sophisticated, interpretable, and flexible patterns of reasoning.”
Graph Technology and AI
Where do Graphs Matter?

- Financial Fraud Detection
- Drug Discovery
- Recommendations
- Customer Segmentation
- Cybersecurity
- Churn Prediction
- Predictive Maintenance
- Search / MDM
Graph Technology and AI - Graph models can provide more accurate predictions with data you already have

- Current data science models ignore network structure and complex relationships
- Graph models can add highly predictive features to existing ML models
Graph Technology and AI
Spark Summit – April 2019 (Neo4j Presentation)

SparkGraph

Neo4j Morpheus

Neo4j Graph Platform

Cypher 9 in Spark to create non-persistent graphs

Cypher 10 over Spark for seamless Neo4j integration

Native Graph Algorithms, Processing, and Storage
Graph Technology and AI
The Steps of Graph Data Science
“Using Neo4j someone from our Orion project found information from the Apollo project that prevented an issue, saving well over two years of work and one million dollars of taxpayer funds.” David Meza, Chief Knowledge Architect NASA 2015
HetioNet is a Knowledge graph integrating over 50 years of biomedical data leveraged to predict new uses for drugs by using the graph topology to create features to predict new links.
Graph Algorithm Feature Engineering

Make use of your existing machine learning pipeline:
• Tabular data from Spark
• Enriched with graph based features from Neo4j
• Combined into a single model building pipeline
Graph Algorithm Feature Engineering
Categories of Features

- **Community Detection**: Detects group clustering or partition options.
- **Centrality / Importance**: Determines the importance of distinct nodes in the network.
- **Pathfinding & Search**: Finds optimal paths or evaluates route availability and quality.
- **Heuristic Link Prediction**: Estimates the likelihood of nodes forming a relationship.
- **Embeddings**: Vectors that capture connectivity or topology.
- **Similarity**: Evaluates how alike nodes are.
Graph Algorithm Feature Engineering
Financial Crime: Detecting Fraud

Connected components to identify disjoint graphs
PageRank to measure influence
Louvain to identify communities
Jaccard to measure account similarity
Graph Embeddings

**Embeddings** transform graphs into a vector, or set of vectors describing topology, connectivity, or attributes of nodes and edges in the graph.

**Vertex embeddings**: describe connectivity of each node.

**Path embeddings**: traversals across the graph.

**Graph embeddings**: encode an entire graph into a single vector.
Graph Embeddings: eBay Explainable reasoning over knowledge graphs for Recommendations
Graph Embeddings: eBay
Explainable reasoning over knowledge graphs for Recommendations
Graph Neural Networks
Example: electron path prediction
Chemicals are graphs - predict a new chemical
Current Research

- Spark Databricks and Neo4j Data Science Training Workshops
- MSc Data Analytics Dissertation supervision. Working with Neo4j Research Team on YELP and Citations datasets and Neo4j posed research questions
- Phd Research UCD Smurfit classifying temporal relationships between events in newsfeed documents using graph analytics
Current Research Data
YELP Dataset Graph Model

156,639 businesses; 4,736,897 reviews of businesses by users; 9,489,337 users; 35,444,850 friend relations
Current Research Data Citation Network Dataset

51,956 papers; 80,299 authors; 140,575 author relationships; 28,706 citation relationships
Current Research Data
Predicting Temporal Relationships in Newsfeeds

Event-relationship graph of newsfeed bbc_20130322_1600