EDM Webinar

Fighting Fraud in Uncertain Economic Times

Live Date: May 21, 2020

Featuring: Dr. Victor Lee, Head of Product Strategy & Developer Relations, TigerGraph

Recording: bit.ly/3bWzkZY

Presentation: bit.ly/36RjQWj

edmcouncil.org **tigergraph.com**

WEBINAR Q&A:

Where do we start and what is the business case?
To start using graph databases and analytics to fight fraud, I would start by using a graph as a supplement to existing data technologies, to represent a Customer 360 and connection-oriented database. In a graph database, if you want to ask the question, "Do THESE entities have THAT attribute in common?", you should model that attribute as a node type, so that you process the question by following connections. Then develop your multi-connection queries which would have been hard to do with a tabular database: Is there a path from A to B? Is there a situation similar to THIS one?

To learn more about using graph analytics and to try a graph database for free, TigerGraph Cloud has a free tier and 20+ different use case Starter Kits. You can learn about TigerGraph Cloud as well as the starter kits at https://www.tigergraph.com/starterkits/. We also have numerous online tutorials, starting with Graph Fundamentals: https://www.tigergraph.com/certification/

Some government agencies have catalogued different types of fraudulent activity - dozens of distinct types, but with some commonalities among them. Can you talk about how you would manage dozens or even hundreds of patterns, especially when the patterns have lots of commonalities among many of them?
Entities, in this case types of fraud, which have things in common sound like a good fit for a tree or graph structure. For example, you might have “Identity Fraud” as a parent type, with subtypes such as “Credit Card Identity Fraud” and “Insurance Identity Fraud.” “Insurance Identity Fraud” might have

another parent called "Insurance Fraud." With multiple parents, you have a graph, not a tree. You then have properties associated with each type, the properties that define or describe that type.

For fraud detection, first select which types are of interest to you. Then consider developing a hierarchical fraud detection system which first looks for a parent type, and then narrows it down to the more precise types. The ontology is just a guide. The decision about which types to focus on really depends on your business needs and the available data to detect fraud.

Decision tree induction – the process of figuring out what is the right structure and questions for your decision tree – can be a machine learning task, which comes prepackaged in many ML libraries. For simple cases, humans can do it just by "staring at the data."

Having both your data and your fraud ontology in a graph enables to take advantage of the information-dense and computational advantages of graph.

The premise of this is terrific. My question is how does a fraud detection approach such as this not become a bad actor in and of itself with massive mining of transactions? Can the government domestic or foreign use it in a spying manner? I ask because this is a big issue, privacy vs security. CCPA, GDPR considerations.

Great question. As a consumer and citizen, you should be informed about the power of data mining – once you have data, what can you learn from it? If your business gathers and stores data about persons, you need to comply with all the legal and ethical considerations. There are also ethics pertaining to how you draw conclusions from that data. As someone in the position of possibly using data mining, statistical methods, and machine learning to make predictions about people – keep the "probabilistic" aspect of the data and learning in mind. Are you certain that the raw data are correct? What is the certainty of the prediction my ML system delivered? And most importantly, what action will I take, based on this probabilistic result? It's one thing to use ML to decide which personalized ads to show someone. It's another thing to deny a person a loan.

What are the pro and cons of using FIBO inside a property graph instead of a rdf store?

FIBO is authored in OWL, which is a knowledge ontology language that is part of the RDF ecosystem. It is basically a classification system. The question is: how do you plan to use FIBO?

It is a great tool to enable transfer and sharing of data between parties, because it is a standard. If you perform a transaction using FIBO names, then it is clear to both parties exact what is being sent and received. As a way of recording information, an RDF store is a solid choice.

Property graphs exceed at computing and analyzing across your graph of data. Typically, the property graph resides entirely within your enterprise, so it doesn't need to fit a universal ontology. FIBO is needed when you are ingesting your data. Once you have the data, it's now categorized. You can then transfer the data you want as needed to a property graph, whose schema has been set up to fit the application you have in mind and perform your analytics and learning.

If you are in a well-defined domain, you can design your property graph schema to match the domain's ontology, and then ingest your data directly into a property graph. Property graphs are just as good as, if not better than, knowledge graphs for storing data with a known ontology like FIBO. This is because property graphs are more streamlined. RDF databases have the overhead of supporting any conceivable input and then performing first-order logical induction on that data.

I am in the banking sector and fraud is clearly front and center and new cases emerge daily. Is graph a new approach or one that is being widely used or rapidly adopted?

Graph has already been adopted by both the largest and the most innovative financial institutions. As the technology and solutions mature, it is becoming easier and easier for institutions to adopt graph approaches. DB-engines.com identified graph as the fastest growing database technology. Four of the top 5 banks in the world are using TigerGraph.

How scalable are such solutions? What has been the adoption in the Fin Services sector?

Scalability was a design goal for TigerGraph from day one. Our database engine uses MPP (massively parallel processing) and distributed system design. MPP is at the microlevel, meaning every operation that can be done in parallel is done in parallel. During a query, each vertex acts like a mini-processor. Distributed design is at the macrolevel, meaning that you can add on servers, and have the data automatically spread across the cluster, as your data grows.

As a result, performance scales almost linearly with the system size: if your data doubles, you probably want to double the number of servers. Your compute times should stay steady.

Does one start in a line-of-business or across LOB's?

Both approaches have been taken by our customers. For any start, you want to minimize risk, see a good return on investment, and use it as a learning case for how to wisely expand to other use cases.

If you start from one LOB, you are focused on solving one problem. This approach is straightforward. The graph solution could either be a supplement or a replacement for an existing system.

The case for starting a graph solution across LOB's is to provide a unified view which was previously not available. If each LOB operates its own data environment, adding a graph application that pulls selected data from each LOB enables corporate-level offices to easily query and analyze across the enterprise. TigerGraph's MultiGraph feature, a graph-oriented approach to access control, ensures that different groups of users can only see the data that they are authorized to see. This approach is low risk because it is purely additive, it is not disrupting any of the LOBs.

What other use cases does linked data and Tiger Graph potentially assist with?

TigerGraph is used by four out the top 5 banks for real-time fraud detection and credit risk assessment. Customers often start their journey with fraud detection in real-time and implement multiple use cases with TigerGraph such as:

- **Entity Resolution** - TigerGraph is used by doing near real-time entity resolution, identifying and linking potential duplicate accounts as they are created. This approach is superior to traditional batch-oriented approach, as it finds duplicate entities or accounts sooner, links them together for customer insights.
    - **Read the TigerGraph's data science mini-guide** - "In-database machine learning for big data entity resolution" to learn how largest banks, payments processors as well as media companies use TigerGraph for the next-generation entity resolution - https://info.tigergraph.com/entity-resolution-mini-guide
    - **Watch the short explainer video of TIgerGraph Cloud starter kit** - "In-database machine learning for big data entity resolution" - https://youtu.be/O1wLxO_xrFQ
    - Start Free with TigerGraph Cloud to use the new starter kit - "In-database machine learning for big data entity resolution"- https://www.tigergraph.com/starterkits/

- **Recommendation Engine** - This is another popular use case after fraud detection and entity resolution. After you have linked and merged the duplicate users, creating a single identity across multiple accounts, you can effectively understand all of the customer behavior, search, browsing and purchase history to determine the best offers for cross-sell and up-sell.

- **Read the Hyper-personalized recommendations solution brief -** https://info.tigergraph.com/hubfs/Hyper-Personalized-Product-Recommendations-Solution-Brief.pdf
- Start Free with TigerGraph Cloud to use the starter kit - “Recommendation Engine 2.0 (Hyper-personalized marketing)”- https://www.tigergraph.com/cloud/

In addition to these, TIgerGraph is used for a wide variety of use cases, including **Customer 360, Network & IT Resource Optimization, Geospatial and Time-Series Analysis, Enterprise Knowledge Graphs** and much more.

- You can discover the solution pages for these use cases here - https://www.tigergraph.com/solutions/

You can start on fraud detection, entity resolution, recommendation engine, customer 360 and all other use cases with starter kits on TigerGraph Cloud - https://www.tigergraph.com/cloud/