FI·CONSULTING

WHITE PAPER: BY ROBERT CHANG

Using Knowledge Graphs for Anti-Money Laundering and Transaction Monitoring

ficonsulting.com

TODAY'S ANTI-MONEY LAUNDERING (AML) AND TRANSACTION MONITORING SYSTEMS NEED TO BE QUICKER AND MORE AGILE TO IDENTIFY INCREASINGLY COMPLEX FRAUDULENT TRANSACTIONS.

Today's anti-money laundering (AML) and transaction monitoring systems need to be quicker and more agile to identify increasingly complex fraudulent transactions. Current AML systems typically contain two steps:

1. Collecting historical customer information and creating a pattern of typical transaction flow (e.g. transaction volume, number of parties involved).

2. Comparing ongoing activity to the typical pattern and flagging any unusual activity.

The weakness of this approach is that considerable time is spent identifying the fraudulent pattern. By the time the fraudulent pattern is identified, sophisticated fraudsters have already adopted more complex fraud schemes. Due to rapid evolution of fraudulent behavior, often layered behind seemingly innocuous transactions, AML models require greater sophistication to remain effective.

Flexible approaches that utilize advanced computational techniques are needed to adapt to changing fraud patterns and to create effective rules for detection.

Current AML and transaction monitoring efforts may be insufficient for the following reasons:

- The global nature of today's financial networks creates complex, high-dimensional, non-linear patterns.
- Rules based approaches do not scale well and produce high false positive rates.
- Fraudulent behavior is deeply hidden behind innocuous behavior due to complicated account layering.
- If transaction monitoring is based on historical patterns of behavior, it will fail to identify a new fraudulent behavior until it is too late to act.

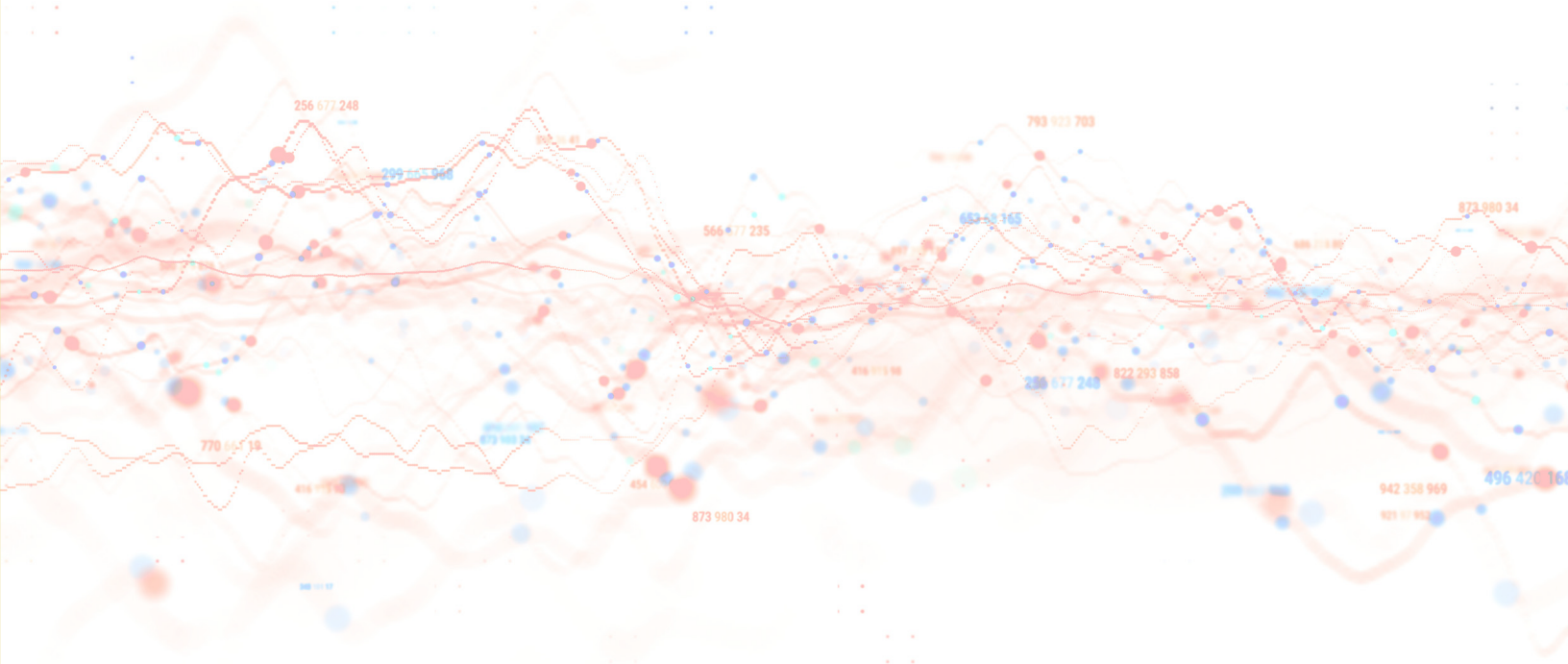
In December of 2018, the Federal Reserve, Federal Deposit Insurance Corporation (FDIC), Financial Crimes Enforcement Network (FinCEN), National Credit Union Administration, and Office of the Comptroller of the Currency (OCC), issued the Joint Statement on Innovative Efforts to Combat Money Laundering and Terrorist Financing. The Joint Statement encourages banks to implement innovative approaches, specifically referencing artificial intelligence (AI). The document states that financial Institutions need to become increasingly sophisticated in their approaches to identifying suspicious activity by building innovative internal financial intelligence units devoted to identifying complex and strategic illicit finance vulnerabilities and threats.

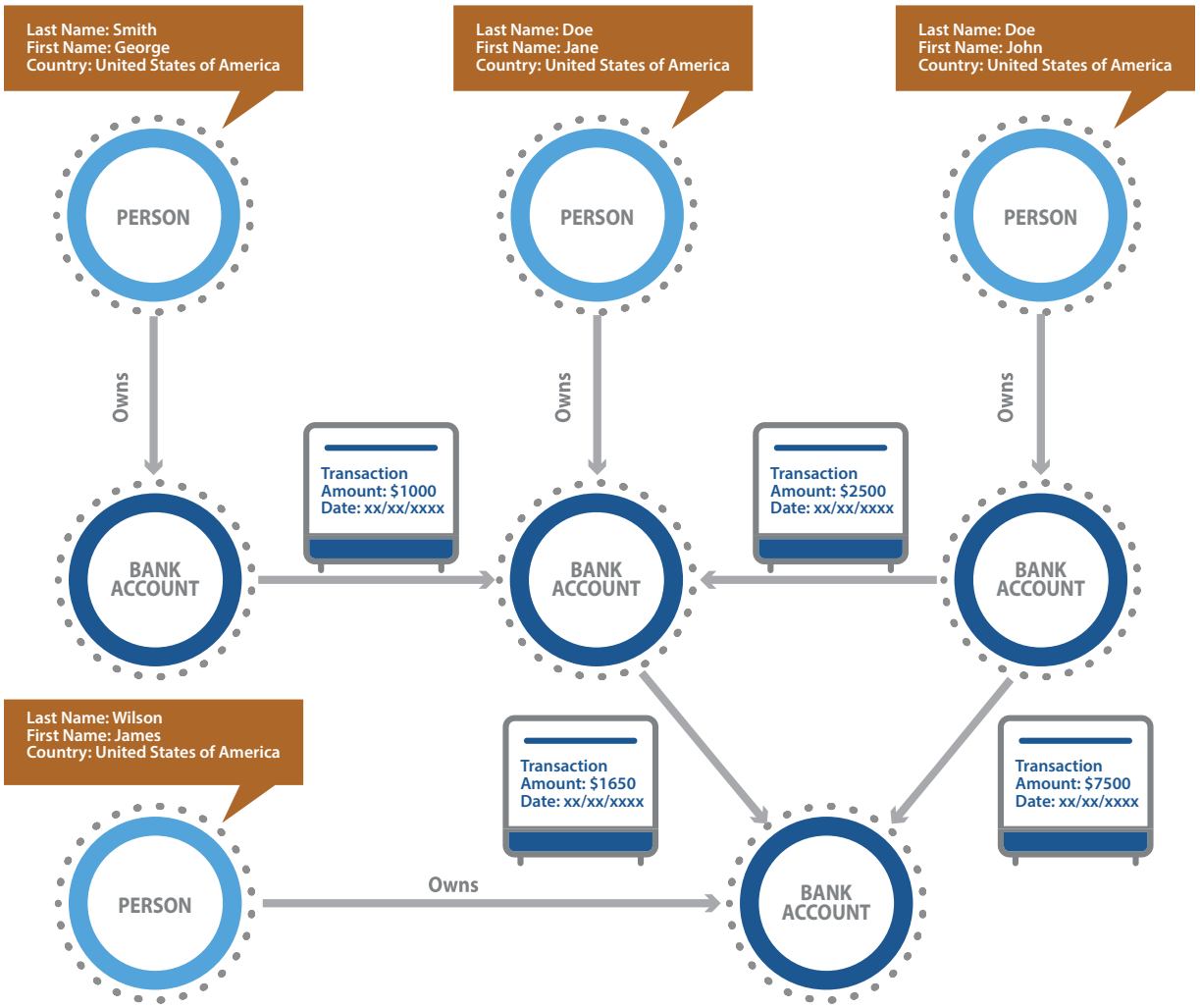
Knowledge Graphs have emerged as an important tool for AML and transaction monitoring. As money laundering involves cash flow relationships between entities, a Knowledge Graph can be used to capture financial transactions. A graph can be formulated where a single account is represented as a vertex and a single transaction between two accounts is represented as an edge. Once the data is captured in a graph database, graph analytics can help to investigate the complex connections between individuals, accounts, companies, and locations. This type of monitoring involves high dimensional mapping of thousands of relationships (edges) between thousands of entities (nodes). Information regarding entities can be collected from standard Know Your Customer (KYC) processes. Information regarding relationships can be collected from observable transactions or Suspicious Activity Reports (SARs).

Knowledge Graph technologies are effective for AML and transaction monitoring due to the following factors:

- They are well-suited for sparse data problems, where actual occurrences of fraud are rare, and do not over-identify false positives.
- They perform classification in non-linear, high-dimensional datasets. Knowledge Graphs are also suitable for time-series transaction data.
- Instead of being rule-based, graphs identify structural patterns and provide a holistic view of customer behavior.

In this blog we demonstrate two graph analytics techniques, clustering and label propagation. Clustering can be used to focus investigation on certain high-risk sectors, while simultaneously reducing focus on low-risk sectors. This provides an efficient allocation of analyst resources and reduces false positives. Label propagation helps find previously unknowable patterns that may have been missed by analysts in the transaction monitoring process, thereby reducing false negatives. Let's look at an example graph to demonstrate the idea.





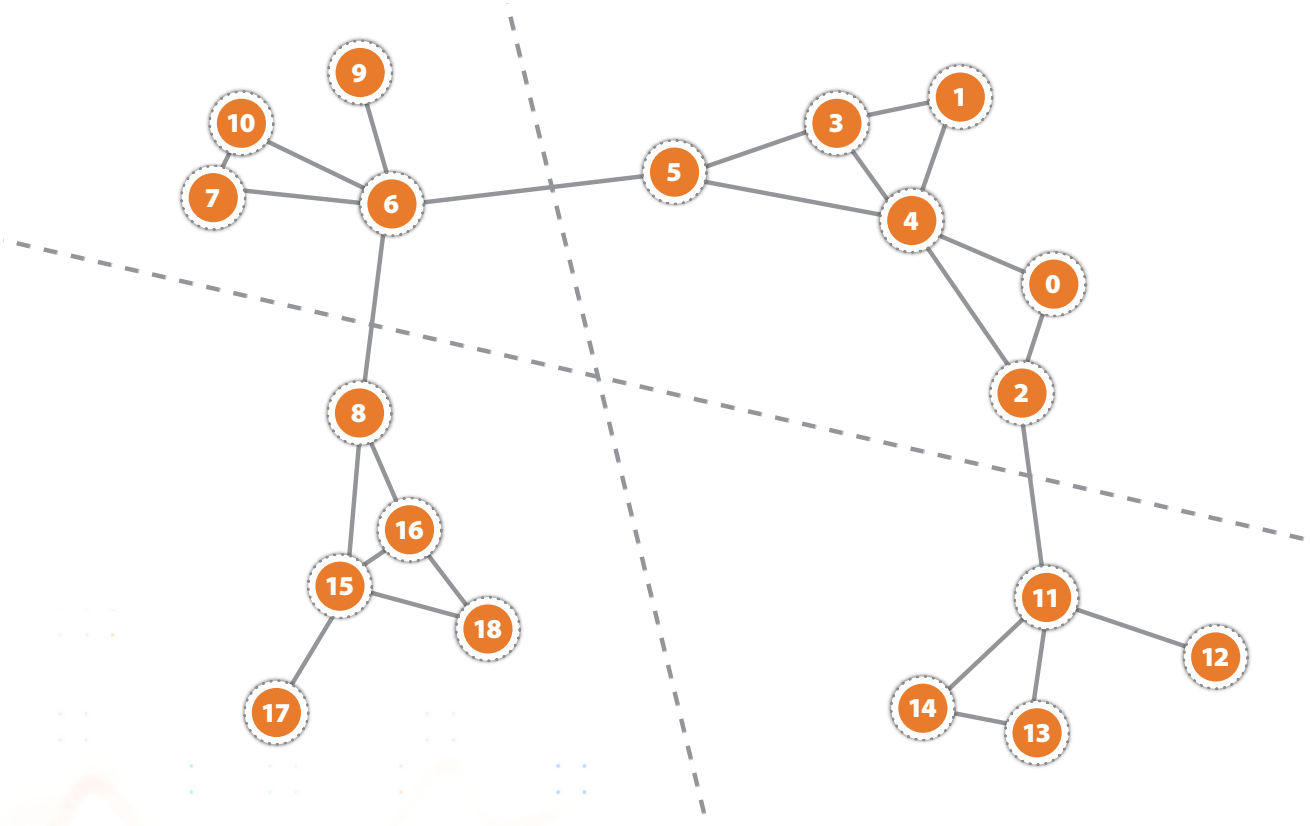
When data is organized in a Knowledge Graph, a key concept is the "connectedness" between nodes. In AML, connectedness can represent a single transaction between two accounts or aggregate transaction volume with a neighboring node over time. Once a graph is constructed, connectedness can reveal relationships between nodes including:

- Closeness – identify which nodes are most important to other nodes within a graph
- Connected components – indication of relationships among groups of nodes
- Community detection – detecting groups that are densely connected internally but loosely connected externally
- Ranking – establishing the trustworthiness of the node relative to the trustworthiness of similar nodes

Each of these analyses play an important part in AML. Graph analytics detects discrepancies and anomalies from typical observed behavior in real-time.

Clustering

Communities of nodes that share many of the same edges should naturally be clustered together. In the case of our example, this would mean different accounts that made transfers to the same account would be likely to be clustered together. To obtain the results in the figure below, we use an algorithm called spectral clustering to partition our graph. Spectral clustering has partitioned our graph into four distinct clusters. As can be seen with accounts #11, #12, #13, & #14, accounts that have transactions with a common distinct account are clustered together. Even though account #12 does not have a direct connection with accounts #13 or #14, it is still part of their cluster due to their similar transactions with account #11. If an instance of fraud is found in a particular cluster, we will know to prioritize reviews for accounts that are members of that cluster. Similarly, an alert found in a cluster that does not have any cases of fraud could be a false positive and classified as low risk. This allows us to quickly identify accounts that require higher scrutiny, even if they are not directly connected to a suspicious account.

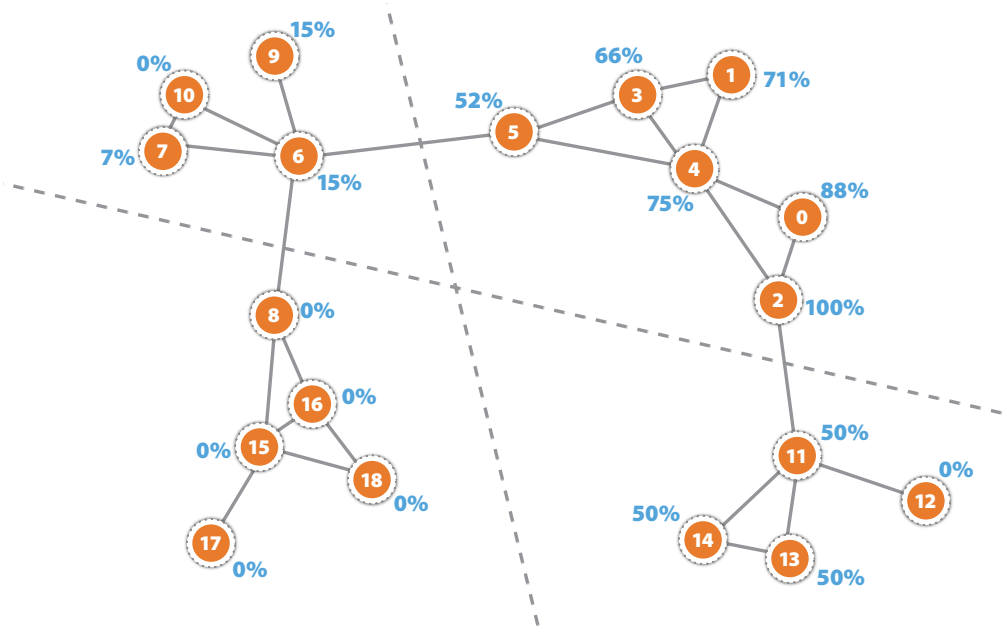


Label Propagation

Label propagation helps find previously unknowable patterns that may have been missed in the transaction monitoring process, thereby reducing false negatives. Transaction monitoring is a quintessential "needle in a haystack" problem, where there is a very small amount of known fraud and very large amount of unknown cases. Label propagation best leverages our sparse known cases of fraud to find common properties among them, and then applies these properties to unknown cases to see if any are similar. Let's look at our example again, but this time in the context of label propagation.

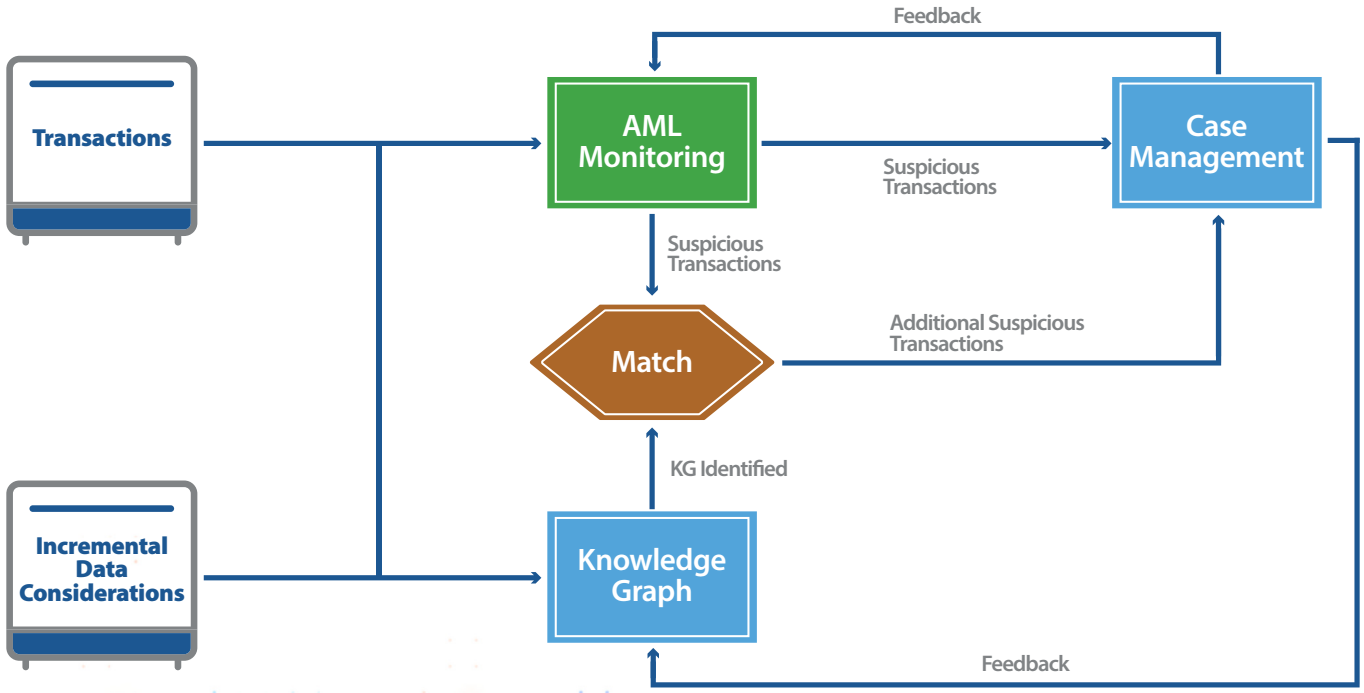
In this example, we will assume that account #2 is a known case of fraud and accounts #8, #10, and #12 are cleared cases that are known to be non-fraudulent. Based on the community structure of the graph we can use label propagation to discover the probability that other accounts are also fraudulent. Label propagation assigns a probability of fraud to each unlabeled node in the graph. For example, accounts #0 and #4 have high likelihoods of being fraudulent due to their shared edges with account #2 and should be prioritized for investigation. Account #11 has a 50% chance of being fraudulent as it shares an edge with account #2, but it also shares an edge with account #12 which is a known non-fraudulent case.

In practice, Knowledge Graphs map thousands of relationships between thousands of entities. In AML transaction monitoring, a node might be a single account or a set of associated accounts. A node could also represent another graph from a previous step in a time series. Traditional relational database systems for fraud detection require complex joins that are difficult to construct. The graph model provides a more flexible schema that accommodate new data as AML models change and evolve. Knowledge Graph platforms such as Stardog and AnzoGraph provide robust graph modeling and querying features in addition to native machine learning capabilities. The result is a platform that can integrate data from across the organization into a common data framework, forming a foundation for AML models and analytics.



Augmenting Existing AML Operational Systems

The Knowledge Graph platforms mentioned can be used in conjunction with existing AML operational systems. The platform can be configured to use the same transaction data inputs as the existing AML Transaction Monitoring system. It can also be used to evaluate incremental data considerations before investing in operational system updates. Suspicious transactions identified by the Knowledge Graph that are not identified by the existing AML system should be routed to the case management system. Feedback from the case management process, e.g., true/false, should be fed back to the Knowledge Graph for on-going monitoring. This feedback can also be used to evolve the existing AML operational system business rules and models.



Appendix: Detailed Explanation of Calculations

A. The clustering and label propagation examples used the following unweighted and undirected graph:

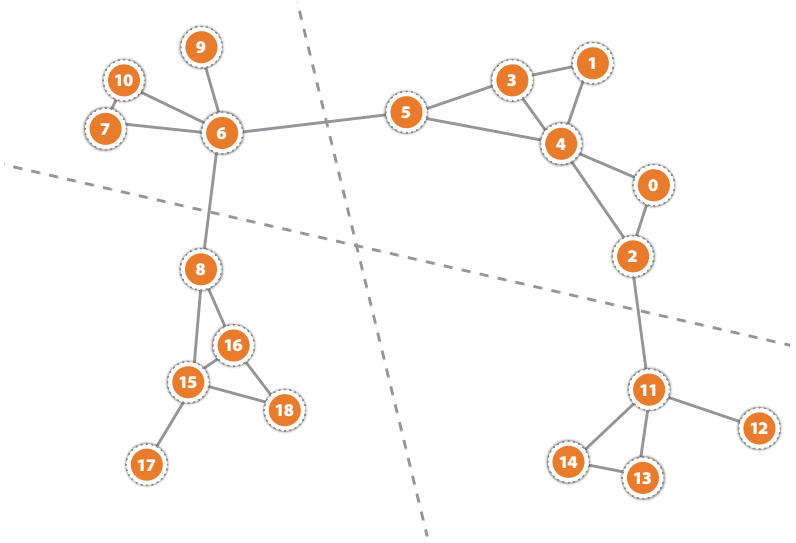
B. Clustering is based off the Eigenvalues and Eigenvectors of the normal Laplacian matrix of a given graph. The Laplacian matrix is the result of two other matrices, the adjacency matrix and the degree matrix. The adjacency matrix is constructed by creating a matrix where the rows and columns indices represent the nodes, and the entries represent an edge between the nodes. What this means is that the adjacency matrix for the example should have a one in the fifth entry of the first row, which will represent the edge between node #0 and node #4 (note how since the example starts at node #0 instead of #1). The degree matrix is a diagonal matrix where the value at the entry (i, i) is the degree of node i. That is to say the entry on the diagonal will be equal to the number of edges that the node has. For our example the first entry in the diagonal for the degree matrix should be equal to two since node #0 has two edges.

Finding the normal Laplacian from these two matrices is simple as the normal Laplacian is simply the degree matrix minus the adjacency matrix. The next step is to find the Eigenvectors and Eigenvectors of the normal Laplacian Matrix. When the Eigenvalues are sorted in ascending order, the second Eigenvalue is called the Fiedler value, and the corresponding vector is the Fiedler vector. The Fiedler value is a measure of how well-connected the graph is and can be used to approximate the minimum graph cuts needed to separate the graph into two connected components. The Fiedler vector also provides information about which side of the cut a node belongs on. The Fielder vector in our example is:

[0.02757, 0.02467, 0.06693, 0.01314, 0.02256, 0.00858, 0.00075, 0.00033, -0.00762, 0.00305, 0.00033, 0.23404, 0.95237, 0.10421, 0.10421, -0.02340, -0.00971, -0.09523, -0.01020]

This tells us that Nodes #0-#14 should be on one side of the divide because they are positive, whereas nodes #15-#18 and node #8 should be on the other side. This would result in two separate connected components. This process could then be repeated until cluster sizes are as small as desired.

A.

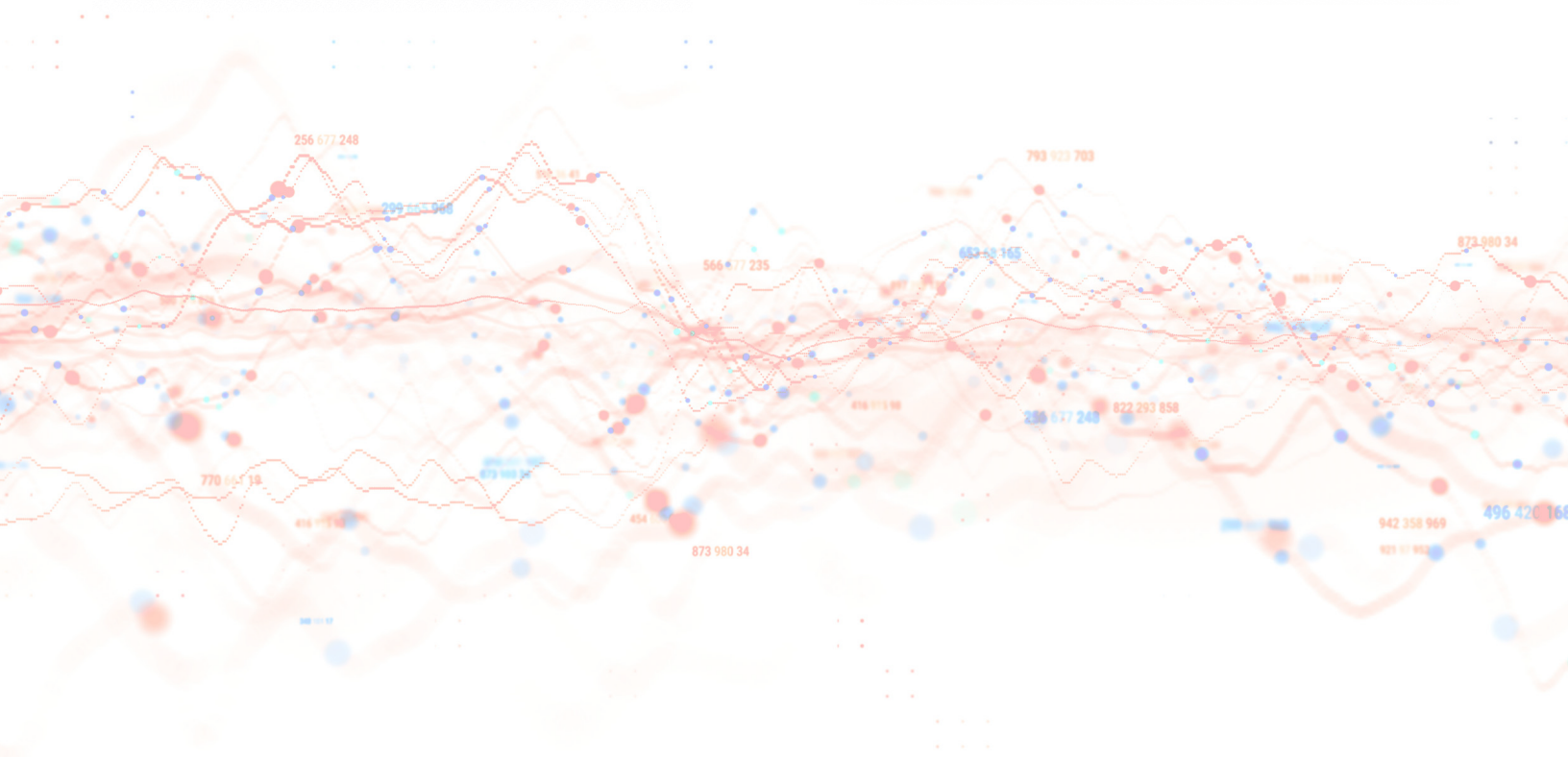


B.

```
[0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0]
```

On the top is the adjacency matrix and on the bottom is the degree matrix

```
[2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2]
```



Appendix: Detailed Explanation of Calculations

C. Label propagation relies on a probabilistic transition matrix, which is similar to an adjacency matrix, but with some minor changes. Nodes that have "known" labels become absorbing states, which means they are empty except for a 1 in the diagonal. After that, each row is normalized, so that all the entries in the row add to one. We create the transition matrix using the same graph as previous, only with the labels of nodes #2, #8, #10, and #12 as "known" absorbing states. This results in the following transition matrix at the right:

This probabilistic transformation matrix tells us the likelihood of traveling to a connected node. To perform label propagation, we then find the long-run equilibrium or steady-state of the probabilistic transformation matrix. This is done by raising the transition matrix to an infinite power. By raising the transition matrix to an infinite power, the transition matrix "converges" to a stable state. In our example, the "steady-state" matrix is at the lower right:

The columns that contain non-zero entries relate to our "known" nodes. Unlabeled nodes are then assigned the label of the "known" node that contains the highest probability. For example, using the first row, node #0 would be assigned the same label as node #2, because column #2 contains row zero's highest probability of 91.03%. Node #6 would be assigned the same label as node #10 because column #10 contains the highest probability in row #6, 49.73%.

C.

```
[ 0   0  .5   0  .5   0   0  0   0  0  0  0   0   0   0   0   0   0   0;
  0   0  .5  .5   0   0   0  0   0  0  0  0   0   0   0   0   0   0   0;
  0   0   1   0   0   0   0  0   0  0  0  0   0   0   0   0   0   0   0;
  0 .33   0   0 .33 .33   0  0   0  0  0  0   0   0   0   0   0   0   0;
 .2  .2  .2  .2   0  .2   0  0   0  0  0  0   0   0   0   0   0   0   0;
  0   0   0 .33 .33   0 .33  0   0  0  0  0   0   0   0   0   0   0   0;
  0   0   0   0   0  .2   0 .2  .2 .2 .2  0   0   0   0   0   0   0   0;
  0   0   0   0   0   0  .5  0   0  0 .5  0   0   0   0   0   0   0   0;
  0   0   0   0   0   0   0  0   1  0  0  0   0   0   0   0   0   0   0;
  0   0   0   0   0   0   1  0   0  0  0  0   0   0   0   0   0   0   0;
  0   0   0   0   0   0   0  0   0  0  1  0   0   0   0   0   0   0   0;
  0   0 .25   0   0   0   0  0   0  0  0  0 .25 .25 .25   0   0   0   0;
  0   0   0   0   0   0   0  0   0  0  0  0   1   0   0   0   0   0   0;
  0   0   0   0   0   0   0  0   0  0  0 .5   0   0  .5   0   0   0   0;
  0   0   0   0   0   0   0  0   0  0  0 .5   0  .5   0   0   0   0   0;
  0   0   0   0   0   0   0  0 .25  0  0  0   0   0   0   0 .25 .25 .25;
  0   0   0   0   0   0   0  0 .33  0  0  0   0   0   0 .33   0   0 .33;
  0   0   0   0   0   0   0  0   0  0  0  0   0   0   0   1   0   0   0;
  0   0   0   0   0   0   0  0   0  0  0  0   0   0   0  .5  .5   0   0];
```

The sum of each row in the probabilistic transformation matrix should be equal to one.

```
[0 0 .9103 0 0 0 0 0 .0327 0 .0490     0 0 0 0 0 0 0 0
 0 0 .8739 0 0 0 0 0 .0446 0 .0669     0 0 0 0 0 0 0 0
 0 0     1 0 0 0 0 0     0 0     0     0 0 0 0 0 0 0 0
 0 0 .7478 0 0 0 0 0 .0892 0 .1339     0 0 0 0 0 0 0 0
 0 0 .8207 0 0 0 0 0 .0654 0 .0981     0 0 0 0 0 0 0 0
 0 0 .5715 0 0 0 0 0 .1604 0 .2406     0 0 0 0 0 0 0 0
 0 0 .1632 0 0 0 0 0 .3315 0 .4973     0 0 0 0 0 0 0 0
 0 0 .0816 0 0 0 0 0 .1657 0 .7486     0 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0     1 0     0     0 0 0 0 0 0 0 0
 0 0 .1632 0 0 0 0 0 .3315 0 .4973     0 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0     0 0     1     0 0 0 0 0 0 0 0
 0 0 .5000 0 0 0 0 0     0 0     0 .5000 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0     0 0     0     1 0 0 0 0 0 0 0
 0 0 .5000 0 0 0 0 0     0 0     0 .5000 0 0 0 0 0 0 0
 0 0 .5000 0 0 0 0 0     0 0     0 .5000 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0 .9888 0     0     0 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0 .9814 0     0     0 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0 .9888 0     0     0 0 0 0 0 0 0 0
 0 0     0 0 0 0 0 0 .9851 0     0     0 0 0 0 0 0 0 0]
```

Author: Robert Chang

Rob is FI Consulting's Modeling & Analytics Domain Leader, managing Data Science teams at clients including the U.S. Department of Health & Human Services, the U.S. Small Business Administration (SBA), Capital One Bank, and the Federal National Mortgage Association (Fannie Mae). Rob leads FI's Graph Practice, implementing graph-based modeling, analytics, and knowledge management solutions for FI's Federal and Commercial clients. Prior to joining FI, Rob was a Financial Engineer and Derivatives Trader for banks and hedge funds in New York, London, Singapore, and Hong Kong. Rob holds a Master's degree in Mathematics of Finance from Columbia University, and a Master's degree in Information Management Systems from the Harvard Extension School.

If you are interested in learning more about how FI Consulting can support your organization, please email contact@ficonsulting.com or call us at 571.255.6900.

FI·CONSULTING

Information. Insight. Impact.
We help clients better manage their **complex** portfolios

1500 Wilson Boulevard | 4th Floor | Arlington, VA 22209 | 571.255.6900 | ficonsulting.com | info@ficonsulting.com