Introduction to Data Science: Network Data

Héctor Corrada Bravo

University of Maryland, College Park, USA

2020-02-24

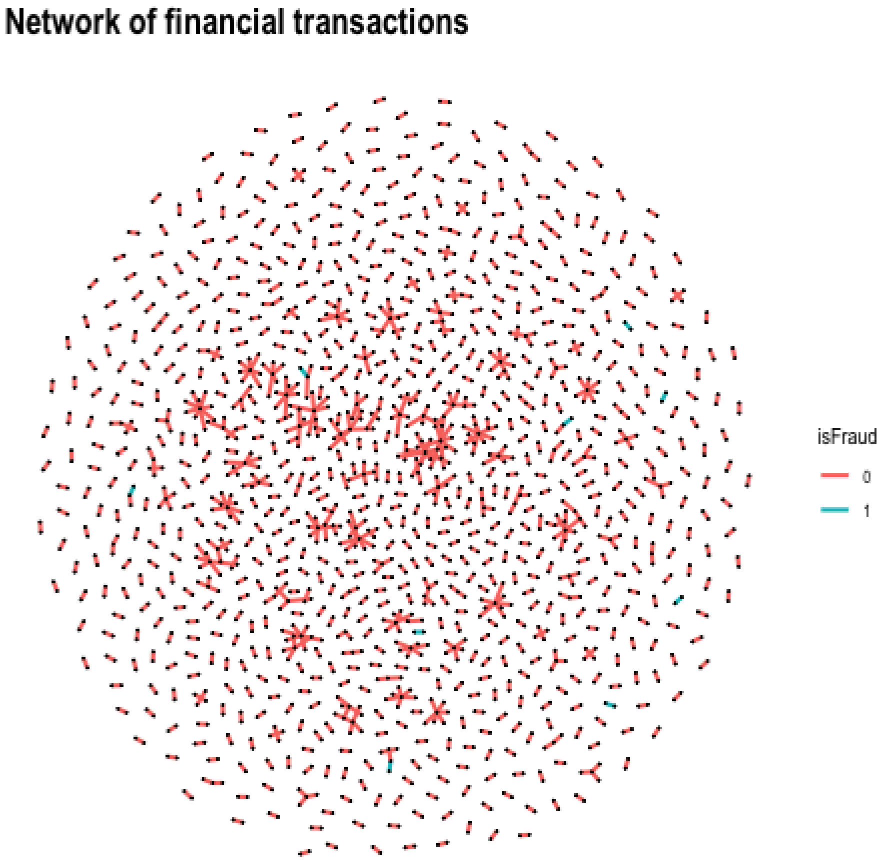
Network Data

In many applications we have data about entities, but also have data about interactions between entities.

Dataset of financial transactions available from Kaggle at https://www.kaggle.com/ntnu-testimon/paysim1.

Some of these transactions are marked as fraudulent.

Network Data



Preliminaries

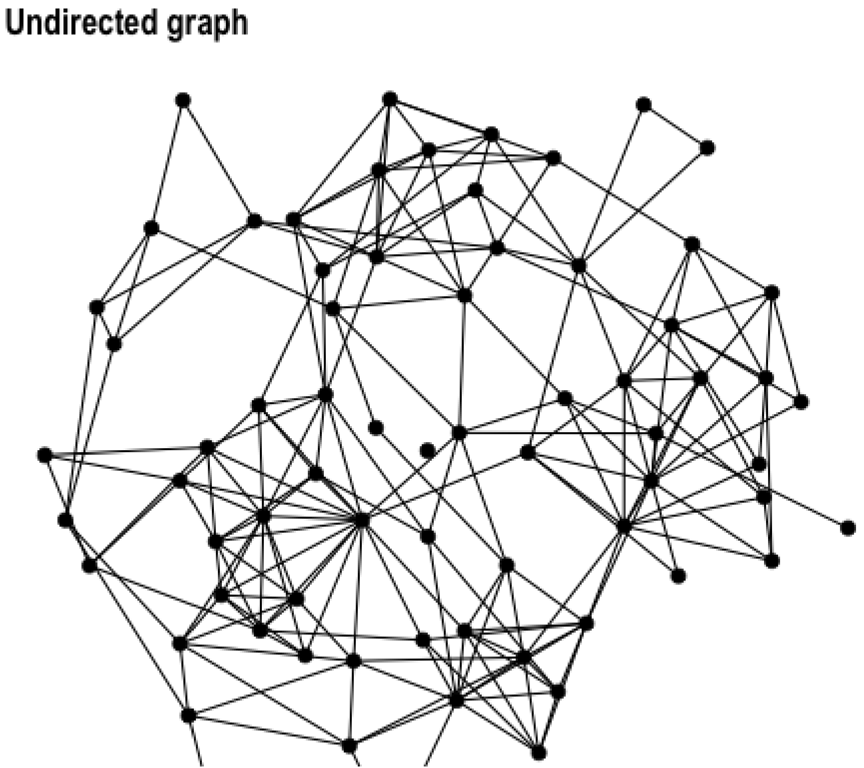
Think about ways to represent data about entities and their interactions.

Mathematically, we use a **Network** as an abstraction of *entities* and their interactions.

We can use a **Graph** as a mathematical representation of this data. In this case, *vertices* represent nodes (entities), and *edges* represent links (interactions).

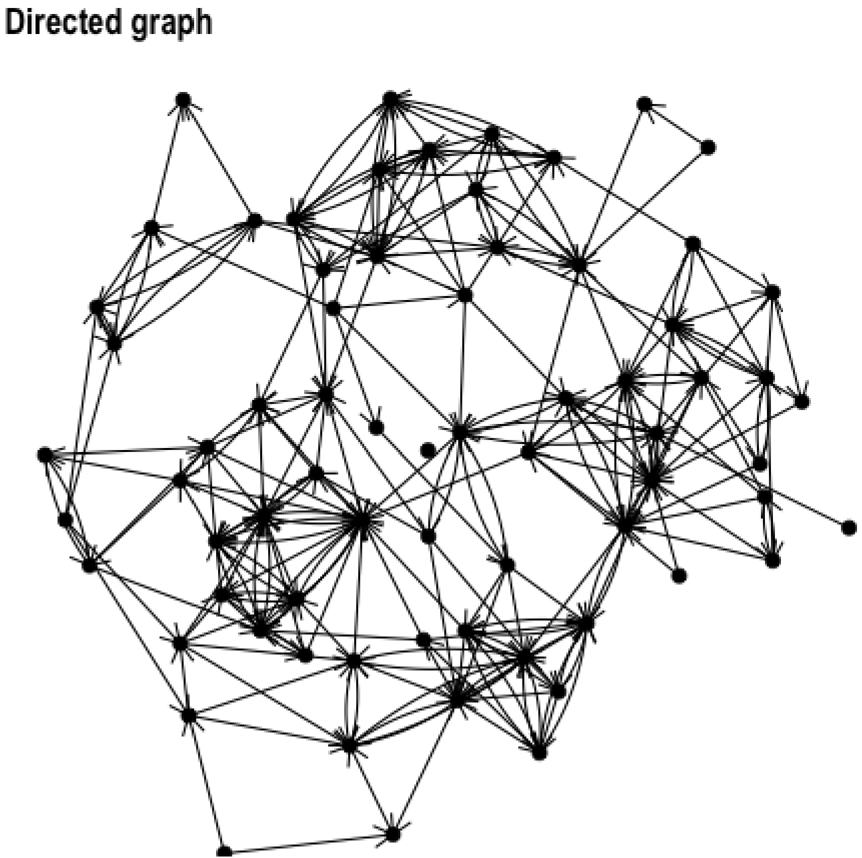
Preliminaries

Here is another graph as an example. In this case edges (or links) do not have directionality.



Preliminaries

We can also represent directional interactions with directed edges.



Preliminaries

In terms of our previous discussion on tidy, rectangular datasets, this is a case where we need to have two distinct tables to represent this data.

- One table represents entities and their attributes:

```
## # A tibble: 1,920 x 2
##    name        node_type
##    <chr>       <chr>
##  1 C746757564  C
##  2 C336400944  C
##  3 C1562533966 C
##  4 C1889457907 C
```

Preliminaries

- Second table to represent edges and their attributes:

```
## # A tibble: 1,101 x 11
##      from    to  step type   amount oldbalanceOrg newbalanceOrig
##     <int> <int> <dbl> <chr>   <dbl>         <dbl>          <dbl>
##  1      1  1102     2 CASH… 8.43e4      7929846.       8014145.
##  2      2  1103     1 PAYM… 2.60e4            0              0
##  3      3  1104     5 PAYM… 2.50e3            0              0
##  4      4  1105     2 PAYM… 9.63e3         6847              0
##  5      5  1106     3 PAYM… 1.53e4         9083              0
##  6      6  1107     1 CASH… 1.68e5        30040.        198492.
##  7      7  1108     2 CASH… 6.31e4      6308037.       6371153.
```

Network-derived attributes

Besides attributes measured for each node, in our example the type of party (Merchant or not for example), we can derive node and edge attributes based on the structure of the network.

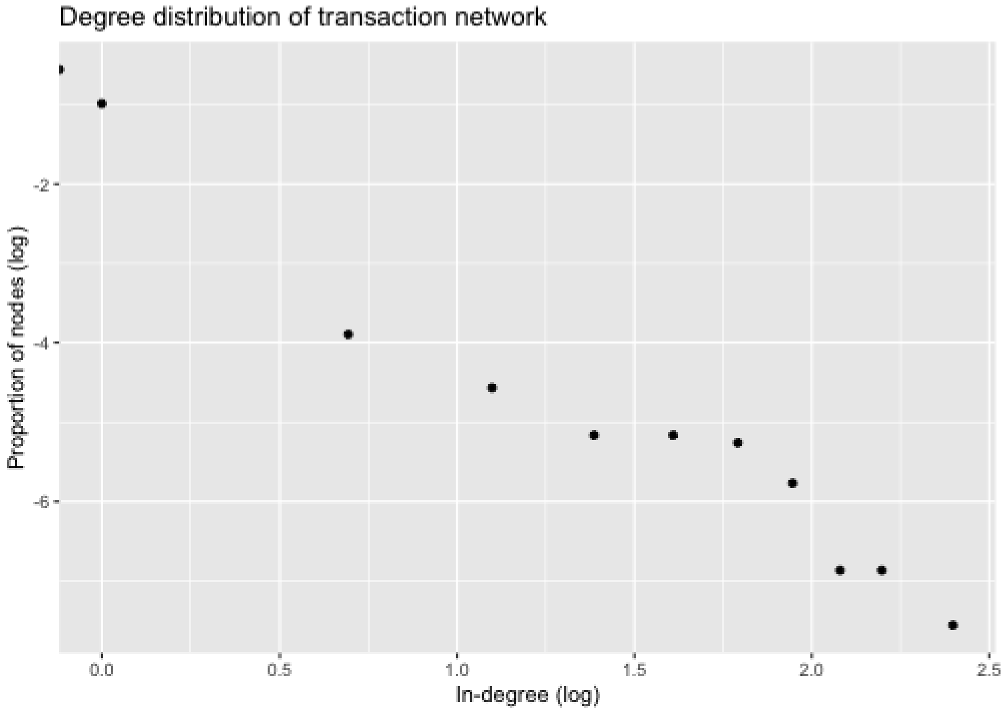
For instance, we can compute the *degree* of a node, that is, the number of edges incident to the node.

Network-derived attributes

```
## # A tibble: 1,920 x 4

##    name        node_type in_degree out_degree

##    <chr>       <chr>         <dbl>      <dbl>

##  1 C746757564  C                 0          1

##  2 C336400944  C                 0          1

##  3 C1562533966 C                 0          1

##  4 C1889457907 C                 0          1

##  5 C1940311161 C                 0          1

##  6 C501036152  C                 0          1

##  7 C1594857799 C                 0          1

##  8 C916100517  C                 0          1

##  9 C1029898472 C                 0          1
```

Network-derived attributes

The distribution of these newly created attributes, e.g., degree, are fundamental analytical tools to characterize networks.



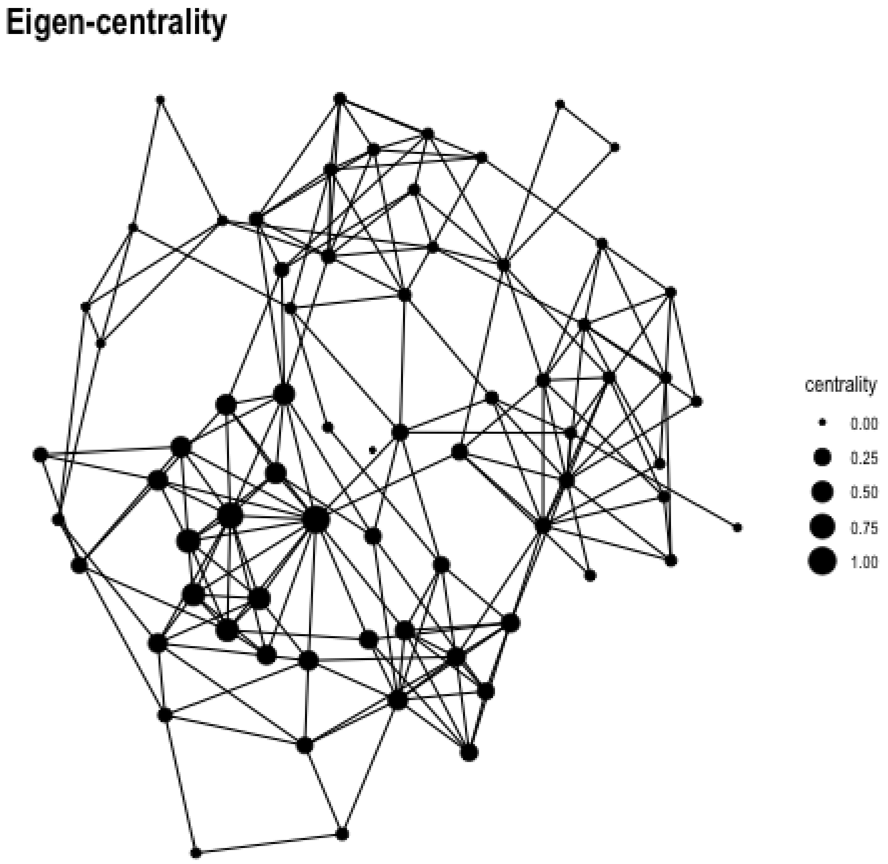
Network-derived attributes

High-degree nodes are *important* to the network since they interact with many other nodes in the network?

It would be useful to know if the nodes they interact with are also *important* nodes.

This is referred to as *centrality*.

Network-derived attributes

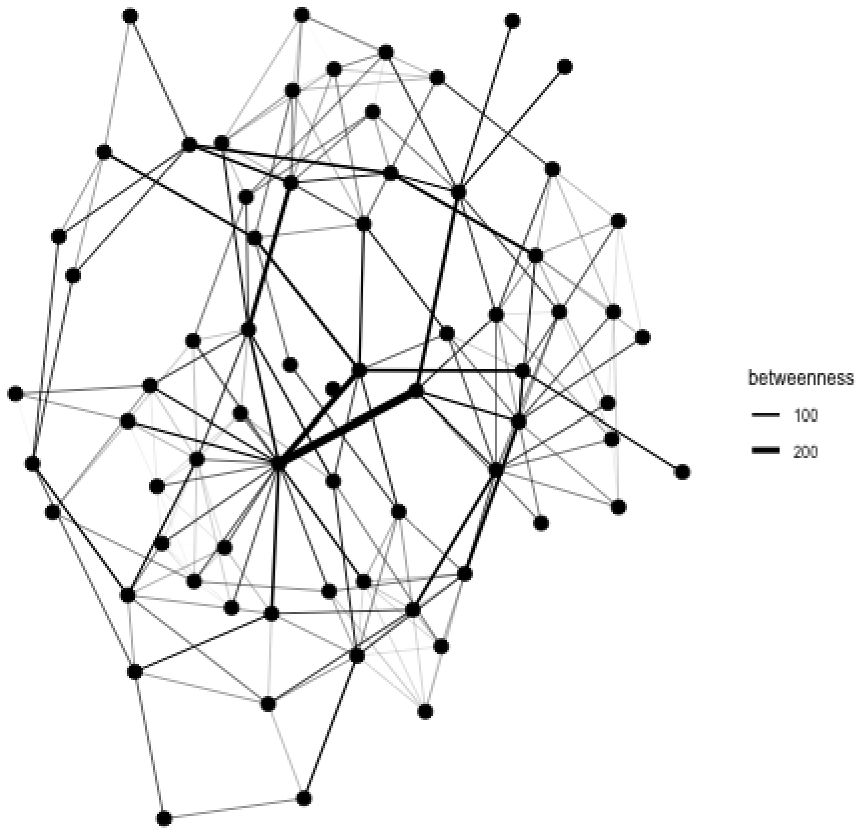


Network-derived attributes

We can similarily think of *important* edges in the network. What are edges that may connect clusters of nodes in the network?

One measure of edge importance is *betweeness*.

Network-derived attributes



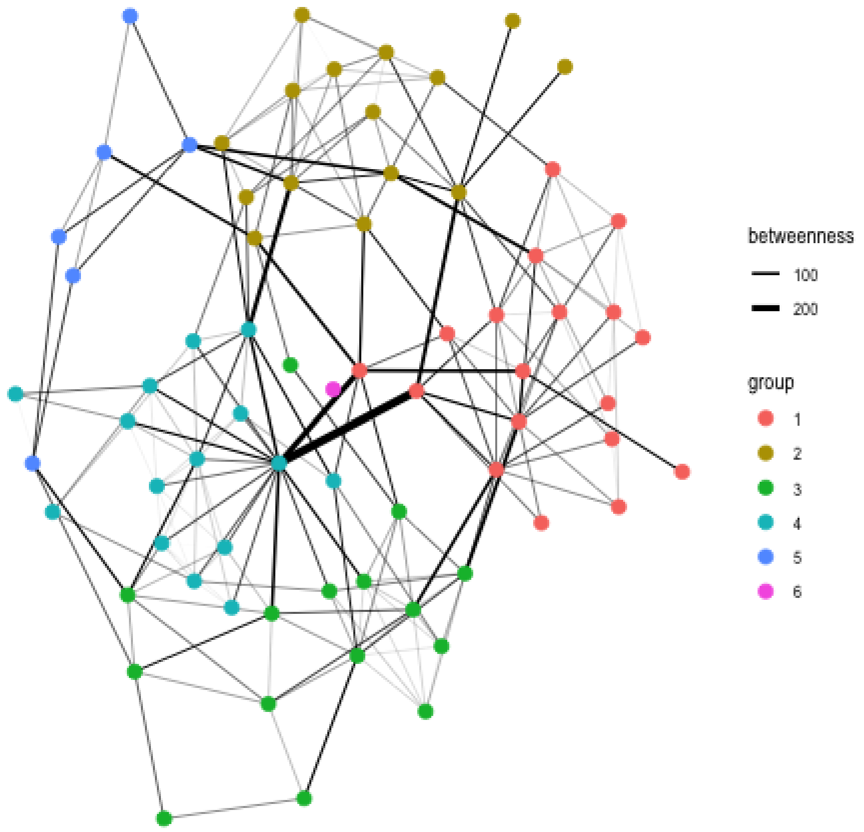
Network-derived attributes

These types of network-derived attributes can in turn be used to understand topological properties of networks.

For instance, we can use betweeness to find *communities* or clusters of nodes in the graph.

The Girvan-Newman Algorithm is a hierarchical method to partition nodes into communities using edge betweenness

Network-derived attributes



Network-derived attributes

Calculating Betweenness

Formally, $\mathrm{betweenness}(e)$: fraction of node pairs (x, y) where shortest path crosses edge e

For each node x, use breadth-first-search to count number of shortest paths through each edge in graph

Sum result across nodes, and divide by two

Network-derived attributes

Resources

There are a number of very useful R and python software tools to represent and manipulate network data.

Cross-language

igraph: http://igraph.org/ Extremely powerful tool for the representation, manipulation and visualization of network data. It underlies many of the R and python network libraries.

Resources

R

In R, the most commonly used packages are:

- `igraph`
- `Rgraphviz`

Newer pacakges use the tidy data paradigm to represent and manipulate networks:

- `tidygraph`
- `ggraph`

Resources

Python

In python, most common tools are:

- `igraph`
- `networkx`