

INFSCI 2415 Information Visualization

Report for Homework 2: Generate Network Visualizations Using D3

Team Member and Contribution:

He, Jiexiao (jih102@pitt.edu)

Song, Chen (chs222@pitt.edu)

Xie, Jingran (jix73@pitt.edu)

Jingran completed the first visualization. Jiexiao completed the second visualization. Chen adjusted the final diagrams and drafted the report. We discussed and designed the visualizations together.

Exploration and Preprocessing of the Dataset:

The dataset for this homework has a graph structure. For nodes (books about politics) there are 4 features: Id, Label (book name), Class (political ideology: l - liberal, n - neutral, c - conservative) and Degree (number of connections). For edges (frequent co-purchasing of books by the same buyers) there are Id of source node and target node.

The dataset is clear and nothing to manipulate. We only change the two csv files into one json file for later usage.

Visualization Design:

Visualization I: A Force Directed Graph Layout with proper visual encodings

(1) Node color: Color the nodes based on the nodes' "Class" field (i.e., all nodes from the same class have the same color). Make the class visually distinguishable from each other. (Hint: Ordinal Scales)

We encode nodes' Class with 3 colors: blue for liberal, green for neutral, and red for conservative. These three colors are similar to the basic colors for screen display and easy to distinguish. More importantly, they are the colors follow the political conventions so it will be easier for the users to understand the information in this visualization.

The annotation of the colors we used is at the top-right corner of the diagram.

b) Node size: Adjust the size of the nodes based on the nodes' "Degree" field. You may try different scaling methods (linear, square root, logarithm, etc.)

We encode nodes' Degree (numbers of connections) with node size. Since the original degree of nodes has a large range, we scale it by first taking a square root. This can shrink the difference of size between each node. And then multiplied by 3 to make sure the small node is visible and interactable.

c) Node label: Show the label of each node based on the node's "Label" field.

The label of node shows that Label field which is the book name. And it will be filtered as we will discuss below.

d) Filtering: Only show node labels with larger Degree (e.g., > 10). How does this new graph compare with the one showing all labels?

We add a slider which can filter nodes by their degrees to different levels (0 - 25). The annotation above the slider suggests the function of the slider and the current filter. Before using the slider all labels will be hidden by default. After using the slider, the label of nodes which survived from the filter will be shown. And all the nodes will remain the same while the filter changes.

This filter can help to find out the number of nodes have connections above a certain number. For example, there are only 2 books have 25 connections (using filter >24) and no book has connections above 25 (using filter >25).

Other:

We also add a mouse hover function to this visualization: when pointing one node, there will be a small window shows the detail information of current node, including book name and degree. The visualization will also highlight the current node and the nodes connect with it by increasing transparency of all other nodes. And when the mouse comes out, all nodes will be return to their original colors. This function can help to identify frequent co-purchasing books of a certain book easily.

Summary of this visualization:

Since the Force Directed Layout naturally shows the clusters, we can see the relationship between nodes' Classes and their clusters in this visualization. Basically conservative books form a cluster and liberal books form another one. The highlight function emphasizes this pattern again: when we highlight all connected books with a conservative book, most of them

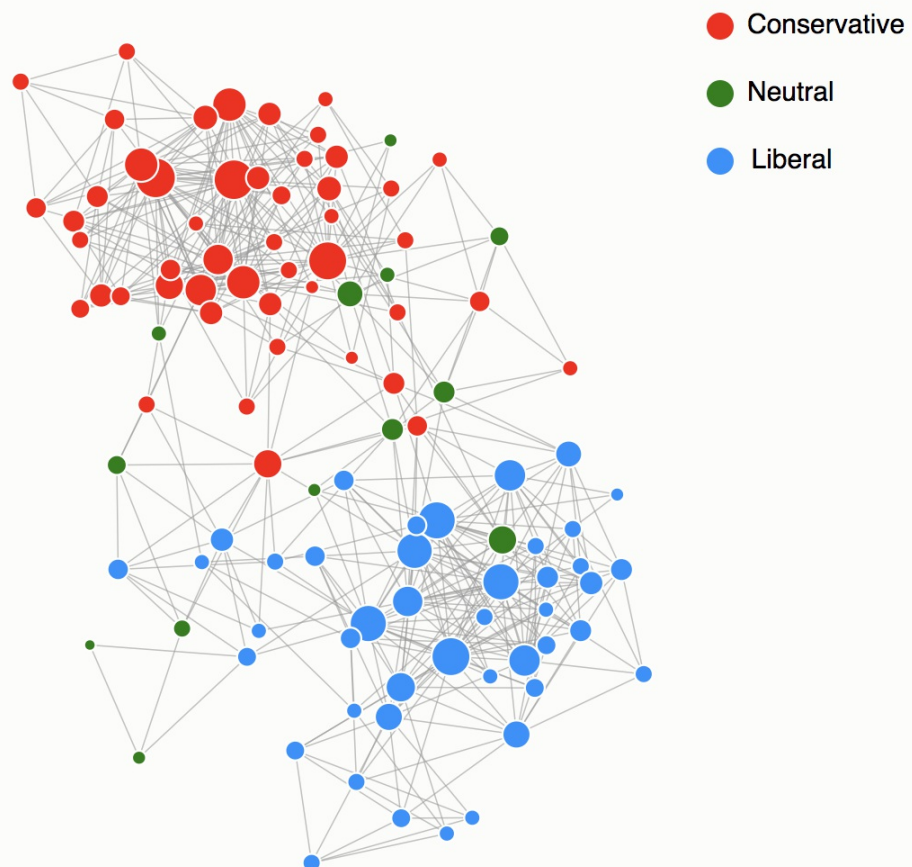
are red, and most of the books connected with a liberal book are blue.

This relationship between nodes' Classes and their clusters tells us there is a potential preference of the users brought different political books: a user brought a conservative book might buy more conservative books, and a user brought a liberal book might buy more liberal books. It is possible that this preference comes from the users' political affinities.

Screenshots:

1. Force-Directed Graph

You can drag slider to change filter

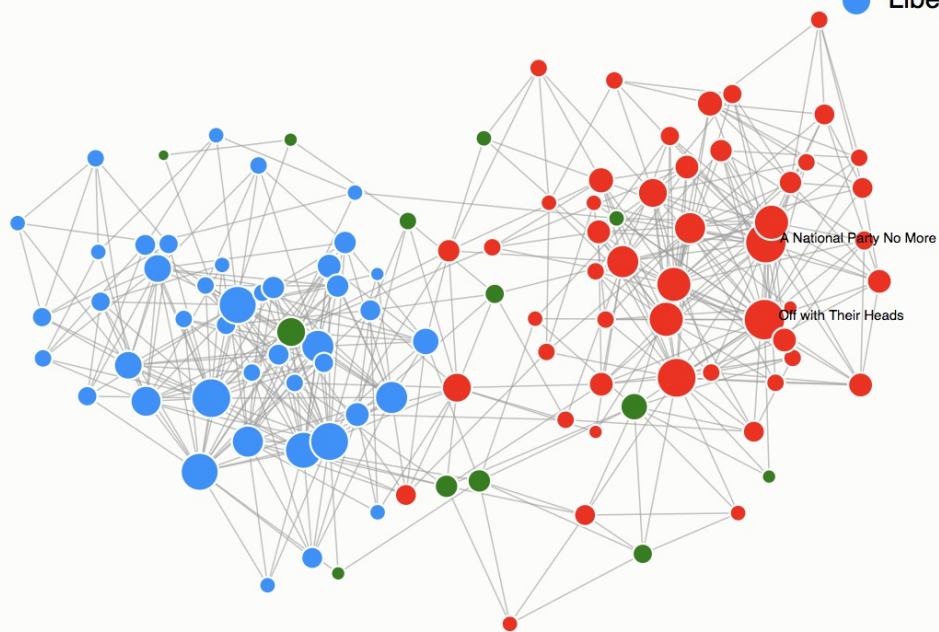


1. Force-Directed Graph

Current filter is degree > 24



- Conservative
- Neutral
- Liberal



1. Force-Directed Graph

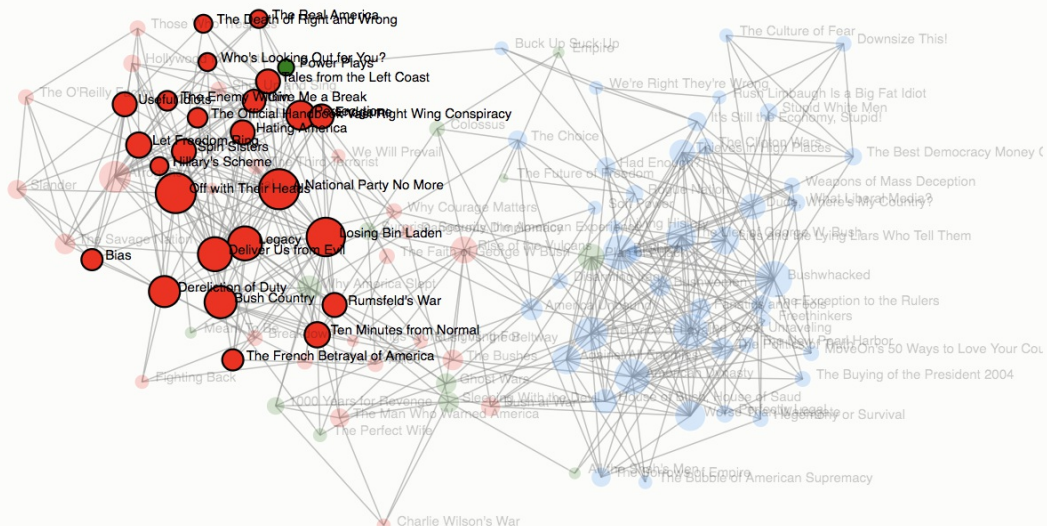
Current filter is degree > 1



● Conservative

● Neutral

● Liberal



Visualization II: Instead of using node-link diagram, explore different options and generate a visualization to show the structure of the same network. The visualization can include multiple charts or mix different types of charts

There are two most efficient ways to visualize graph structures: node-link diagram and adjacency matrix diagram. Since we already created a node-link diagram can show the details of nodes in section I, here we choose to create an adjacency matrix visualization and show some details of the edges. We found a relationship between nodes' Classes and their clusters in the first visualization, so we also expect to see this relationship in our adjacency matrix diagram.

In the adjacency matrix view, the row and column index of the matrix show the Label field of nodes, which are book names. The texts of indices are colored by red, blue and green which is the same with node-link view.

The color of each grid shows the existence and type of each connection as follow:

- white: no connection,
- red: c-c connection,
- blue: l-l,
- green: n-n,
- purple: c-l,
- orange: c-n,
- light blue: l-n.

Since we colored nodes with red for c, blue for l and green for n in both two visualizations, it will be easy to understand that the colors of grids/edges in this visualization are just the mix of nodes' colors. For users who still feel not familiar with the colors, we also have annotation for the color usage at the right side of the diagram.

Another function of this visualization is that we add a scroll list to sort all the nodes by different ways: by their names (alphabetical order), by degrees of nodes and by classes of nodes.

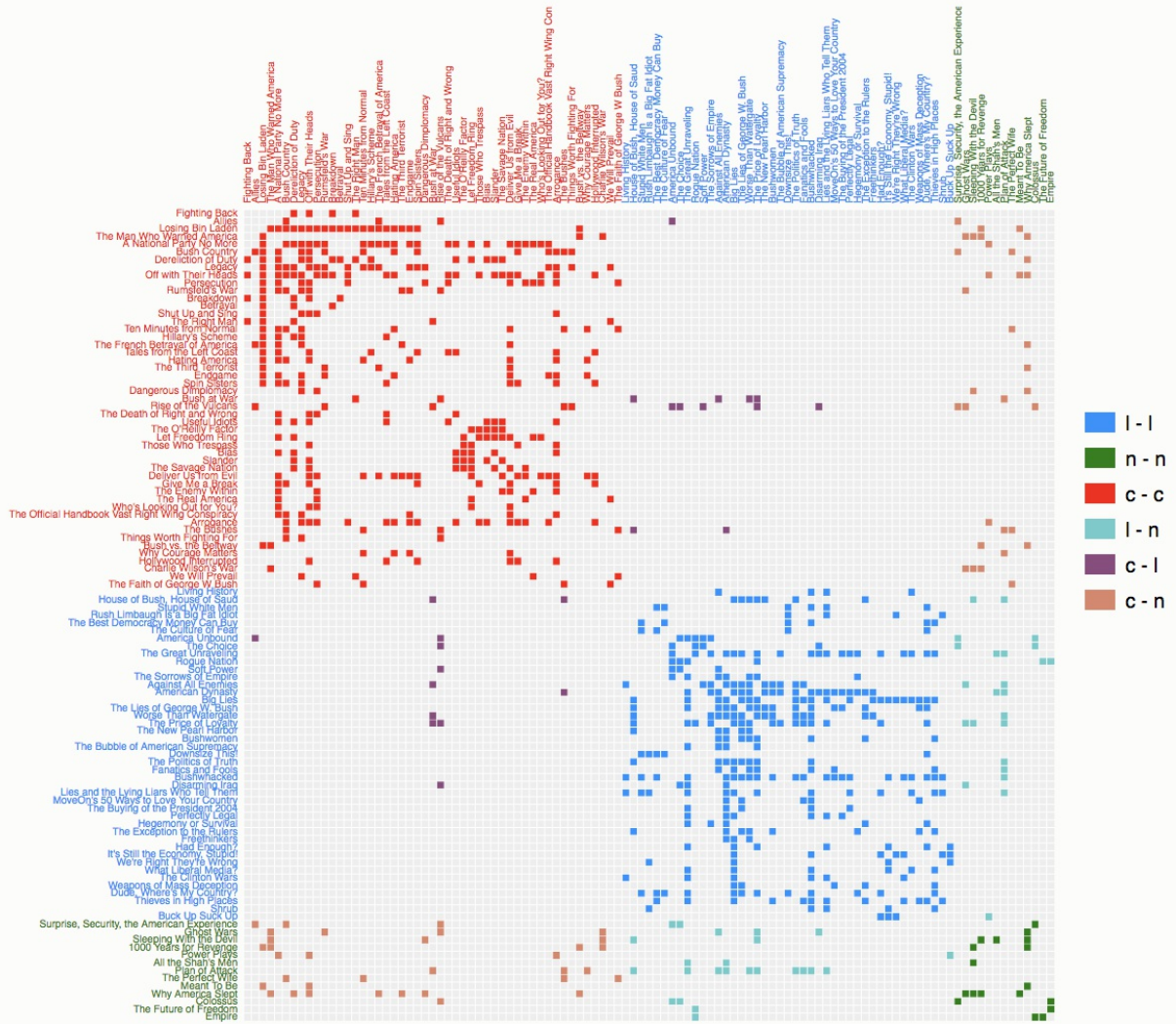
After sorting by classes, it is easy to see that most edges are c-c or l-l. We also have some n-n, n-c and n-l, but only a few c-l connections. This suggests again that people have a preference to buy similar political books.

We also add a mouse hover function in this visualization: when the mouse moves to an exist connection (a not-white grid), the two corresponding nodes indices will be highlighted by increasing their font size. This function will help user to locate any two connected nodes based on the edge between them.

Screenshots:

2. Adjacency Matrix View

Order: **By Class**



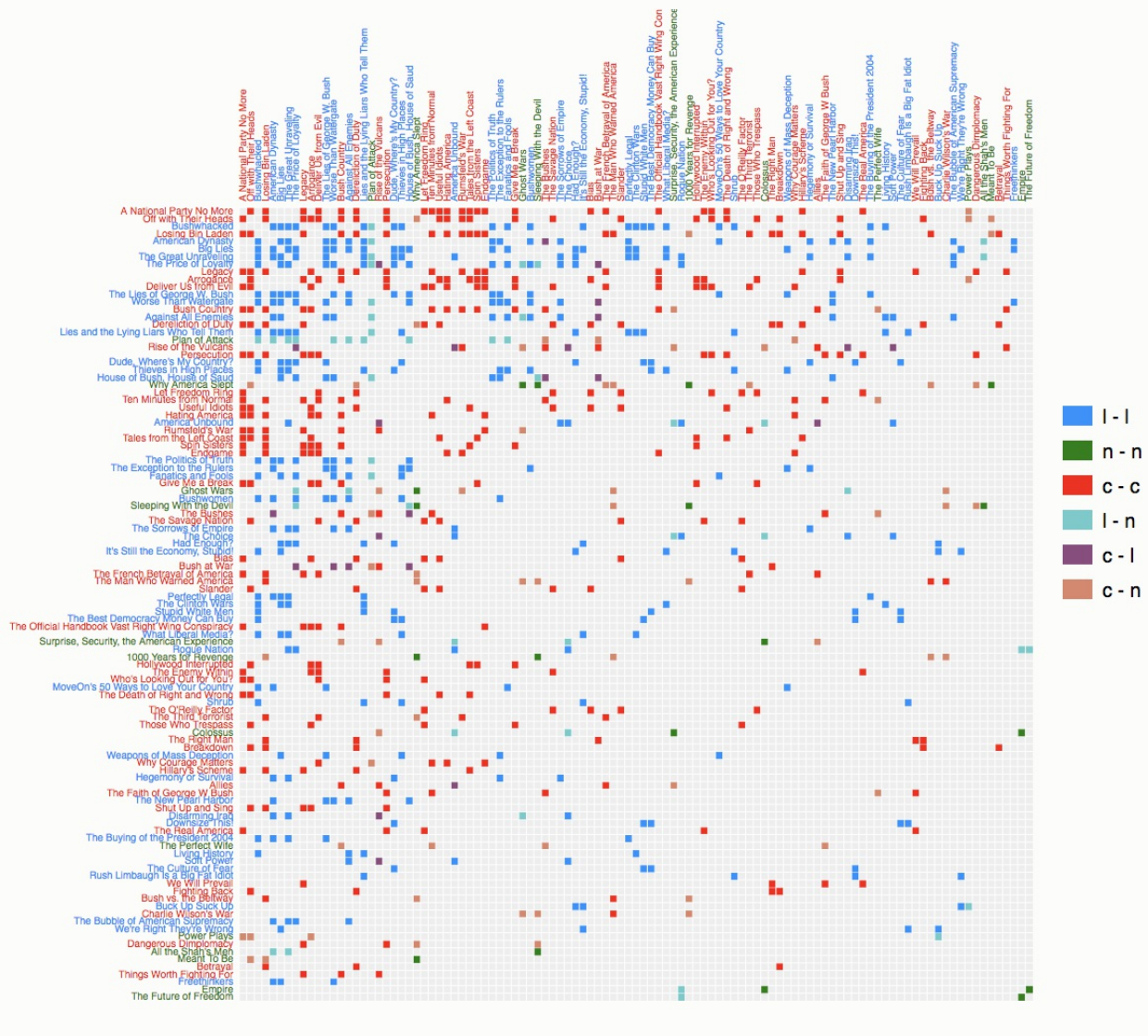
2. Adjacency Matrix View

Order: By Class



- I - I
- n - n
- c - c
- I - n
- c - I
- c - n

Order: By Number of Links ▾



1. Force-Directed Graph, Mike Bostock, October 15, 2017,
<https://bl.ocks.org/mbostock/4062045>.
2. Les Misérables Co-occurrence, Mike Bostock, April 10, 2012,
<https://bost.ocks.org/mike/miserables/>.
3. VI7: Node-link diagram and Adjacency matrix, Hung Do, March 18, 2016,
<http://bl.ocks.org/hungvietdo/7f8df0bcd7fba7e531e6>.