Build-Your-Own Venmo Network Visualization

Final Project for 6.859, Spring 2021

Kiran Gite Shane Weisberg

Master of Business Analytics
Sloan School of Management
Massachusetts Institute of Technology

ABSTRACT

We present our final project for 6.859, Interactive Data Visualization, for Spring 2021: *Build-your-own Venmo Transaction Network*. We use data scraped from Venmo's public API to construct a network of historical transactions that users of our visualization may construct from the bottom up. We hope users will enjoy exploring this network and pause to think about data privacy and the state of their personal data on Venmo and similar platforms. The visualization can be found at `https://6859-sp21.github.io/final-project-byo_venmo_transaction_network/`

1 INTRODUCTION

For our final project for 6.859, we aimed to achieve 3 goals. First, we wanted to spend our time working on an intriguing problem with interesting data and real world applications. Second, we wanted to be able to use this data to put together an interactive visualization that would provide an enjoyable and novel experience for users while also being instructional about the issue at hand. Finally, we wanted to improve our d3 skills by attacking a visualization type with which we had no experience.

This paper will present the visualization we devised to achieve all of those goals, which we have titled *Build-Your-Own Venmo Network*. The visualization can be found at `https://6859-sp21.github.io/final-project-byo_venmo_transaction_network/`. Venmo is an online peer-to-peer payment platform that allows users to send money easily to friends and strangers, but it also has elements of a social network. Users must input a description for their payment before sending or requesting money. The recipient of every transaction, along with the message, will be public unless the user initiating the payment opts to make this information private. The recipient of the transaction does not have input into whether the data is public or not. The amount of the transaction is always private.

Venmo has come under fire many times for its lack of data privacy, and while the company has made changes to improve these issues, the company's attitude towards payment data remains that payments should be social and public. This is clear from the current version of the application. Figure 1 shows a screenshot of the public tab of one of our Venmo feeds, which appears upon opening the app. The public feed tab contains the full first and last names and transaction messages of Venmo users who are strangers and whom we have no connection with whatsoever. If this many transaction details are so easily accessible to us just by making a Venmo account, it is clear that data privacy is still not as pressing a matter as it should be.

Venmo's lack of data privacy was apparent to us when we found a large dataset of historical Venmo transactions available to download from GitHub. This data was scraped from Venmo's public API by Dan Salmon [7], who had the goal of demonstrating the blatant invasions of privacy that publicly available Venmo transaction data allows. Our goal for this project was to incorporate this data into an interactive visualization that would allow users to more easily grasp this point than would a massive data dump in tabular format. We will briefly discuss previous work in this area in Section 2.

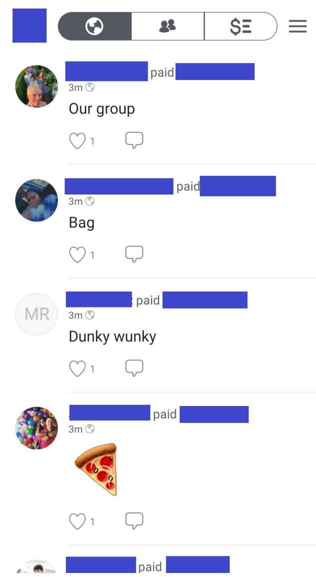

Figure 1: The public tab of one of our Venmo feeds, containing the full names and transaction messages of strangers

The details of the dataset will be discussed in Section 3; like most social networks, we felt this problem lends itself cleanly to a network visualization, with the Venmo users as nodes and transactions as edges. D3 is known for its powerful network visualization functionality, so we took this as an opportunity to try something new and exciting. Specifically, we wanted to enable users to build a network from the bottom up by directly interacting with the nodes. We will go into further detail about the visualization's functionality in Section 4 and the results of our work in Section 5. Finally, we provide a discussion about what we have learned from this process about data privacy and what we hope to impart to potential users in Section 6.

Terminology Note: To remove ambiguity, in this paper we will refer to Venmo users in our dataset as 'actors' and users who are interacting with our webpage as 'users.'

2 RELATED WORK

There has been some study of Venmo as a social network as opposed to just a payment application. Acker and Murthy (2018) [1] discuss how providing users with transaction feeds and messages builds social practices and communication. The authors propose that users may be reclaiming control over their data by adding emojis and messages to payments, treating their payments like posts on social media. The authors also mention that temporal analysis of Venmo payments can be used to understand typical rhythms of payment (for example, rent is usually due on the 1st of every month), and also analyze how typical rhythms are being shifted by modern payment technology (rent payments may be transferred between roommates throughout the month). Users may use payment messages and emojis to describe their payments, or they may be using certain messages and emojis to obfuscate payments made for illicit behavior. The use of Venmo messages to cover up payments for illegal services has recently made the news, with Florida GOP Representative Matt Gaetz using Venmo to transfer money to sex trafficker Joel Greenberg in 2018, who subsequently paid three women, with the messages on the transaction reading ”Tuition” and ”School” [6].

Related work on Venmo in the data visualization sphere has also focused on data privacy. Designer and researcher Hang Do Thi Duc created an interactive data visualization in 2018 called ”Public by Default” which displayed insights from over 200 million Venmo transactions scraped from the public Venmo API [3]. Do Thi Duc shared the minute details she was able to learn about Venmo users' lives, anonymizing the data but highlighting the highly invasive details uncovered. Do Thi Duc's visualization was able to convey important points about the visibility of Venmo data while still respecting the privacy of individuals in the data. The report and visualization were covered widely by news media, and the subsequent scrutiny of Venmo may have led Venmo to limit access to their public API. However, the default privacy setting on Venmo transactions is still "public."

Other research has taken the same approach of generating invasive insights from public Venmo data in order to convince Venmo users that they should not complete public transactions. Yao et al (2018) developed a location inference graph algorithm that is able to predict the top 5 locations of Venmo users with 90% accuracy [4].

Finally, computer science students at Stanford University used the same dataset as we are using to perform network analysis for a class within the Stanford Network Analysis Project [8]. However, they performed community detection using clustering algorithms, while we are taking a data visualization approach.

3 DATA

Data used for this project was obtained from a GitHub repository created by Dan Salmon [7] and consists of all publicly available Venmo transactions scraped from Venmo's public API in certain months between July 2018 and February 2019. In total, this amounts to over 7 million transactions, each represented in a single row of data. To allow the dataset to be hosted on the class GitHub without using the GitHub LFS extension, we filtered out all actors who made fewer than eight transactions. This left us with over 165K transactions.

The d3 network packages expect network data in a very specific format, so the first step was to process this data and convert it into a JSON format while capturing important information about the users and transactions. In this preprocessing step, we count the number of transactions performed by each actor and the number of adjacencies (actors with whom at least one transaction was performed). We also obtain each actor's ID, username, and photo url. In a highly personal dataset like this one, privacy concerns must be at the forefront; we discuss this at length in Section 6. For each pair of connected actors, we record the identities of each party, the number of transactions completed between the two parties, the date and time of each transaction, and all messages sent in either direction. No data on the amount of each transaction is available. All of this data is then converted to JSON format to be loaded into our webpage.

4 METHODS

4.1 Network visualization

As mentioned in a previous section, this problem and data lends itself neatly to a network structure, so we wanted the network visualization to feature front-and-center on our webpage. As we know, there are many important decisions required to make high-quality network visualizations, and these design choices will be discussed here.

Importantly, we decided very early on that it would be impractical and uninformative to ever display the full network on the screen. First, rendering that many objects and calculating forces between them would crash a browser (we can say this from experience), and additionally, it would make for an extremely cluttered and messy visualization where users would have a difficult time extracting information.

As such, we decided to have users build the network from the ground up. Sample views of our network at different stages of construction can be seen in Figure 5. Upon opening the webpage, users are prompted to select one of a set of pre-selected actors to display on the screen before beginning the simulation. These actors were selected as they had a varying number of transactions and neighbors included in the dataset. The selected actor and their immediate neighbors will be displayed on the screen. The rest of the network is not loaded; thus we avoid computational and visual difficulties. Users can then built out the network in several ways. Users can click on any node to add that node's neighbors to the visualization. Additionally, users can input any Venmo username into the search bar at the top of the page. If that username is present in the dataset, it will be added to the visualization. In this way, users can add nodes to the network that are not connected to the initial selection. Lastly, users can input a message string or emojis (which we implemented using an external package called fgEmojiPicker) into a second search bar and this will add all transactions that contain that string to the network along with the nodes on either end of that transaction [9]. As the network grows, users can control it by zooming and panning and hover over nodes and edges for more information. Additionally, as the network grows, the plots at the bottom of the page will update automatically. We will discuss these plots further later in this section.

Beyond the bottom-up structure, we had to decide how to encode additional information in this network. On Venmo, transactions can go two ways: you can pay or receive money. We considered using directed edges to incorporate this additional information, but ended up deciding against it to minimize clutter. Even in the small networks we are displaying, picking out individual edges can be difficult, so we did not want to add additional lines. We chose to encode the number of transactions between actors as the weight of the edge between them. We opted to use a logarithmic scale to size these edges due to the variation in transaction counts. We also decided to color the nodes by their degree. Degree is generally easy to see in networks; however, ours is different in that there is no guarantee that all of a node's edges will be displayed on the screen. Therefore, we think this encoding channel is helpful in guiding users who are interacting with the network. Users know that clicking on a colored node may reveal additional nodes and edges while a black node has no further information to reveal. Additionally, we added functionality so that a node turns slightly transparent once it has been clicked on and its neighbors added to the network. This is useful as it clearly shows the user which nodes have yet to be clicked on and fully explored.

Lastly, we decided to use D3's default force-directed algorithm [2] to position nodes. In this algorithm, nodes repel each other with a strength determined by a given charge value while edges pull nodes together. Additionally, we add a central force to keep all nodes relatively close together for easy viewing. This algorithm generally does a good job of revealing network structure with highly connected nodes and pods falling towards the center of the network and other, less-connected nodes being pushed towards the periphery.

Interactions that can can be performed on the network include clicking nodes to show their neighbors (as mentioned above), dragging nodes to a fixed position for network repositioning, and hovering over nodes and edges to see tooltips. The dragging functionality was important for us because it provides a fallback if D3's force layout with our supplied parameters is unable to produce a meaningful network visualization. If users have the option to drag nodes to better positions, we do not have to leave everything up to the randomness of the force layout, and there is still a way for users to engage with the visualization if the generated layout is suboptimal. We believe the tooltips that appear on hover for nodes and edges add credibility to our design, as users can see the exact numerical values behind our node color and edge width encodings. An important component of the node tooltip is the corresponding actor's Venmo account picture. We wanted to drive home the point that an actor's image is publicly available to all Venmo actors and beyond, but still wanted to respect the privacy of the actors in our dataset. Our compromise was to add a gray filter that transitions over an actor's image as a tooltip appears. Users cannot discern exactly what is contained in the pictures, but they are still made aware that the pictures exist and are available.

4.2 Other interactive plots

We included four interactive plots to highlight trends that are not immediately observable from the network visualization. All four plots update dynamically with the username, transaction date, and message filters present in the visualization. One of the interactive plots is shown in Figure 2 and helps the viewer observe temporal transaction trends. We chose line encodings in order to capture transactions over time, as we thought it was a common encoding for trend charts over time and thus would be straightforward for viewers. In terms of interaction, we included tooltips displaying precise transaction counts when the viewer hovers over a data point on the chart.

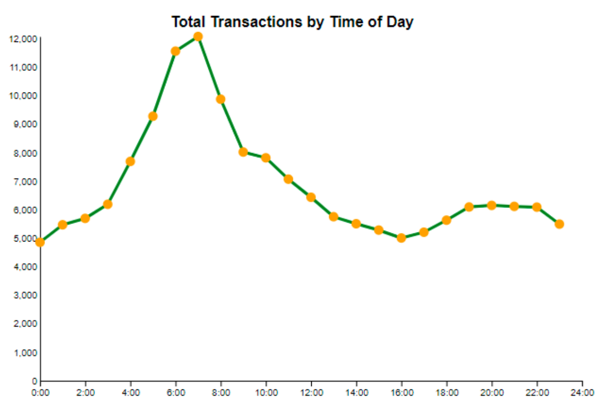

Figure 2: Plot of total transactions by time of day for the entire network. Plots will update automatically as users build/filter the network.

The third interactive plot is a vertical bar chart showing a count of transactions per actor and is shown in Figure 3. We decided to limit the chart to show at most 40 actors to avoid the bars becoming too narrow. We already encode the number of transactions between two actors using edge width in the network visualization, but we believe users may find it valuable to see an aggregation of this information grouped by actor. We chose to encode the number of transactions using bar height due to the straightforwardness of the encoding. Upon hovering over a bar, a user can see a tooltip with the actor's username and transaction count. Upon clicking on a bar, the bar's outline turns green and the outline of the corresponding node in the network visualization also turns green. This allows users to connect the bar chart with the network visualization.

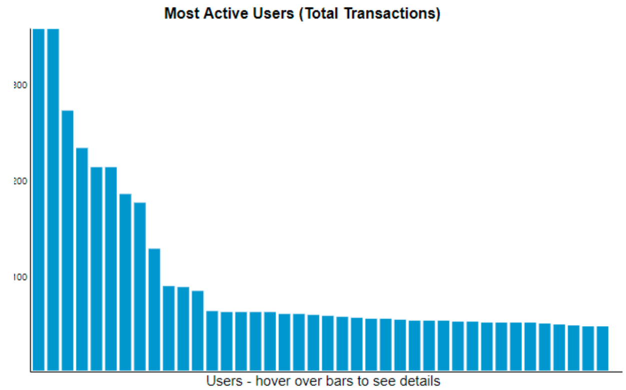

Figure 3: Plot of total transactions for 40 most active Venmo actors in the full network. Plot will update automatically as users build/filter the network.

The final interactive plot is a bubble chart displaying emojis used in the messages between actors in the current network. To create this visualization, we used the external package GraphemeSplitter to split a message into letters and emojis, since a single emoji can contain up to four characters [5]. We then calculated the frequency of each individual emoji in the network. Each emoji is one element of the bubble chart, and the emojis are sized by their frequency, so more frequently used emojis appear larger. We chose this bubble visualization after pivoting away from a bar chart since emojis are lighthearted and playful. We thought that a less standard and academic visualization would convey this playfulness and help the users engage with the visualization. The interactive elements of this plot are the tooltips that appear when hovering over an emoji, displaying the count of this emoji across all transactions in the current network.

Figure 4: Bubble plot of emoji usage for the full network. Plot will update automatically as users build/filter the network.

5 RESULTS

We believe that the bottom up approach to network building we described gives users a more intimate connection with this network and a greater opportunity to explore it as it grows. As an example, the top portion of Figure 5 below shows one of the default configurations that we provide for users. The bottom portion shows that network after expanding the network by clicking on nodes to add neighbors.

This bottom network is still an extremely small subset of our total data, but even so would be unwieldy and confusing to show users on its own. But building this network node by node allows users the time to watch it grow, explore the new nodes and edges using the tooltips, and get a more holistic view of these larger networks.

One case study that illustrates how our visualizations address our problem is the case of a user who wants to learn more about their own Venmo network. The user searches for their username with the search bar and their node appears, along with the nodes of all other actors with whom they have had a transaction, and all the edges representing those transactions. The user is able to hover over all the nodes and edges and recognize the usernames of their friends and connections. The user can then click on an another actor's node to see to what degree that actor is connected within their friend group, as well as other communities that the actor could be part of. Through expanding this network through clicking and username searching, the user hopefully realizes that they can learn personal information about acquaintances or strangers, such as where they went on vacation, who they went out for drinks with, and who their roommates are just based on the messages contained in their transactions. The user then reflects on their own network and realizes that they should set their Venmo transactions to be private or semi-private (Venmo "friends" can still view their transactions).

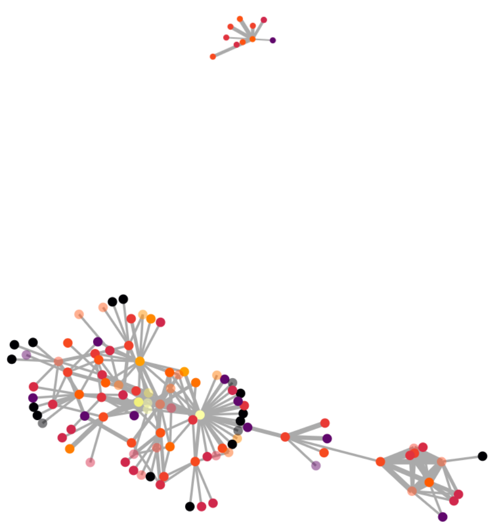

Figure 5: Our Venmo User network at two stages during a user's exploration.

6 Discussion

As discussed previously, our motivation for using this dataset for this project was to enable reflection and discussion about data privacy. This issue has been the subject of much public focus recently, with companies like Facebook and Google in the spotlight. Even so, for many people, this discourse lacks a personal aspect. Most people know their data is being collected and used, but not much more than that. Venmo requires users to opt out of public transactions, and most users likely do not take the extra step to do so. Why is this the default for a platform built around sensitive financial transactions? Venmo wants to market itself as a social platform, and people enjoy seeing custom messages from their friends. But many people certainly do not know that this data is all made publicly available to the world on Venmo's API. The authors of this paper certainly did not prior to this exercise. We believe this visualization, even if a user's own data is not in the network due to data or storage limitations or their own privacy settings, will help users see the personal nature of this data and encourage them to think critically about these policies and practices. The key aspect that makes this visualization personally relevant is the ability to see actors' usernames and censored pictures; otherwise, users might not feel the impact of data privacy concerns as strongly.

Of course, as the handlers of this sensitive data, we needed to make sure that we were not doing exactly the opposite of what we were advocating for by further circulating actors' private information without their consent. As mentioned, this data is already publicly available on the app, the API, and Dan Salmon's GitHub, but were we to share this visualization broadly, we would be making it available in a much more accessible format. There is a delicate balance here; we want to emphasize the sensitive nature of the data while not violating the actors' privacy without their consent. As such, we made the choice to hide the most sensitive pieces of information: we use actors' public usernames rather than their full names and hide their profile images from view. Since all of this information and more is readily available on any user's Venmo feed, and Venmo even displays the transactions of strangers to its users to encourage social posting, we believe our final visualization does not overstep any boundaries of privacy.

7 Future Work

Going forward, if we were to continue developing this project, the most important next step would be to use the full transaction dataset, rather than a subset. Currently, we are only including actors who made eight or more transactions in the available time span (note: these actors may still be displayed as having fewer than eight transactions as any transaction between an actor in the network and an actor not included in the network will not be counted). As we are never displaying the full extent of the network (this crashes the webpage, even at this size), the limited data may not seem like an issue, but we expect the first thing many users will do upon reaching our page is to input their own username into the network; most names are currently unlikely to appear in the subset of the network we are using, though.

Beyond this, we could experiment further with different network formats and test out different ways of displaying the full transaction history between actors beyond a tooltip.

We would also like to dig deeper into network analytics and graph theory to be able to report more statistics about the network. We could include different measures of centrality beyond just the number of adjacent nodes to examine which actors are most prominent in the network. Specifically, eigenvalue centrality (actors who are adjacent to other central actors must themselves be central) and betweenness centrality (actors who appear on many shortest paths between nodes must be central) lend themselves well to this dataset. And on the topic of shortest paths, implementing some sort of search algorithm to find the distance between nodes and display the shortest paths between actors would be a cool opportunity for interactivity as well.

Acknowledgments

The authors wish to thank Professor Arvind Satyanarayan and the entire 6.859 teaching staff for constructing and running a fun and informative class this term.

References

[1] Amelia Acker and Dhiraj Murthy. Venmo: Understanding mobile payments as social media; https://dl.acm.org/doi/abs/10.1145/3217804.3217892.

[2] D3. D3 wiki: force layout; https://d3-wiki.readthedocs.io/zh_CN/master/Force-Layout/.

[3] Hang Do Thi Duc. Public by default; https://22-8miles.com/public-by-default/.

[4] Yao et al. Beware of what you share: Inferring user locations in venmo; https://ieeexplore.ieee.org/abstract/document/8372910.

[5] orling.

[6] Jose Pagliery and Roger Sollenberger. Gaetz paid accused sex trafficker, who then venmo'd teen; https://www.thedailybeast.com/gaetz-paid-accused-sex-trafficker-who-then-venmod-teen.

[7] Dan Salmon. https://github.com/sa7mon/venmo-data.

[8] Aarush Selvan, Druv Kedia, and Rohan Sampath. Motifs and community detection in the venmo transaction graph; http://snap.stanford.edu/class/cs224w-2019/project/26425095.pdf.

[9] woody180 and RahulChand028. fgemojipicker.js; https://github.com/woody180/vanilla-javascript-emoji-picker.