Identifying Bottlenecks in Specialty Referral Networks

Brehm, Jacob
Goergen Institute for Data Science
University of Rochester
Rochester, NY 14627
jbrehm@@ur.rochester.edu

Swar, Sayan
Goergen Institute for Data Science
University of Rochester
Rochester, NY 14627
sswar@ur.rochester.edu

Abstract

At the University of Rochester Medical Center (UR Medicine), referrals typically originate in primary care and are sent out to the appropriate internal and external specialty practices. Ideally, these referrals are sent out upon request and processed immediately, without too much hassle for the medical staff or the patient. Unfortunately, it is not uncommon for bottlenecks to occur in the referral pipeline, leading to extra work for the medical team and added uncertainty for the patient. In this project, our goal was to be able to identify where these bottlenecks are occurring—i.e., which departments and specialties are least efficient, and if there are any referral pathways that are convoluted in particular. We achieved this objective through the use of an intuitive network visualization, embedded in a web application that provides additional functionality and interactability.

1 Introduction

At UR Medicine, most referrals originate in primary care and are sent out to practices that are both internal and external to the medical center. Ideally, these referrals are dealt with efficiently, i.e., they are dispatched upon request and then processed by the receiving practice immediately, without too much lag time or hassle for the medical staff or the patient. However, it is unfortunately not uncommon for bottlenecks to occur in the specialty referral pipeline, leading to extra work for the teams at each practice in addition to added uncertainty for the patient that the referral is being made for.

The reasoning for the existence of bottlenecks in the speciality referral pipeline can be quite varied. For instance, a certain speciality may be referred to much more commonly than other departments, which means that the speciality practice must be quite efficient in approving or denying the referral. Similarly, certain specialties may be more selective when allowing referrals, independent of how many referrals they actually receive. For example, at UR Medicine, the rheumatology department is much more selective when accepting referrals than the physical therapy department.[1]

There is also the notion that some departments may simply have a better baseline efficiency than others in regards to making referral decisions and pushing them through the system. For instance, according to Dr. Fear and Dr. Zelenka of UR Medicine, there is evidence that many referrals get delayed or never even scheduled in the first place; there is also evidence that some referrals have not even been looked at for months. This inefficiency could be attributed to many different things, such as understaffing, mistakes, referrals being a low priority, communication issues between the referring department and the specialty practice, etc.

[1]Based on a verbal statement by Dr. Zelenka.

There are many other things to consider in the context of the referral network as well. For example, how should we weigh one referral against another—is it viable to weigh referrals by some metric of urgency, should they be prioritized by the time they entered the system, or should there be some more complex mixture of both? Additionally, there are certain instances where a department logically makes more referrals to other departments than are being referred to them; this leads to the question: what should be considered a bottleneck?

In this project, our goal was to be able to identify where these bottlenecks are occurring—i.e., which departments and specialties are least efficient, and if there are any referral pathways that are convoluted in particular. We aimed to arrive at a conclusion using measurable data and evidence, and to be able to visualize the referral network in such a way that would illustrate and support our claim.

We also hope this project opens the door to other interesting discoveries about the data, which could include quantifiable metrics such as: the departments that initiate the most referrals, the specialties that receive the most referrals, which departments are most and least efficient, which pathways are particularly convoluted, and if there are any clusters or communities of practices that seem to most commonly exchange referrals.

The ultimate goal, beyond the scope of this project itself, is to provide a foundation for performing further analysis and gaining more granular insights about the specialty referral network at UR Medicine.

1.1 Problem Statement

In essence, our goal with this project is:

> Can we visualize the bottlenecks that occur in the specialty referral network so that opportunities for improvement can be further explored?

2 Data Set

The data was shown and explained to us in an initial meeting with Dr. Fear and Dr. Zelenka. In its current form, the data is stored in a SQL database, and contains information about each referral. We were also informed that the data originally included patient information, but these columns were redacted in order to retain the privacy of each patient. A snapshot of the data that was taken during the demonstration is shown in Figure 1.

The dataset contains approximately 400,000 rows, which provides plenty of data for the purposes of this project. Each row of the data corresponds to a single referral that is being made from a

	REFERRING_DEPARTMENT	REFERRING_SPECIALTY	REFERRED_TO_SPECIALTY	REFERRAL_CLASS	REFERRAL_NETWORK_STATUS	IS_LEAKED_YN	EXTERNAL_REFERRAL_REASON	CLOSE_REASON	FIRST_APPT_ASSIGNED_DT
9...)	LeRoy Medical Associates	Family Medicine	DERMATOLOGY	Outgoing	Out of Network	Y	Patient Preference	External: Pt to Schedule	
9...)	Clinton Medical Associates	Internal Medicine	SLEEP MEDICINE	Internal	NULL	N	NULL	Duplicate Referral	
9...)	Olsan Medical Group	Internal Medicine	OTOLARYNGOLOGY	Internal	NULL	N	NULL	All Visits Complete	
9...)	UR Medicine Primary Care Marion	Family Medicine	ALLERGY/IMMUNOLOGY/RHEUMAT...	Outgoing	Out of Network	Y	Authorizing Provider Preference	Appointment Scheduled	
9...)	UR Medicine Primary Care - Bushnells Basin	Family Medicine	PEDIATRIC NEUROLOGY	Internal	URMFG	N	NULL	All Visits Complete	
9...)	Clinton Medical Associates	Internal Medicine	PHYSICAL / OCCUPATIONAL THERA...	Outgoing	NULL	Y	NULL	Patient Deferred	
9...)	Rochester Internal Medicine Associates	Internal Medicine	NEUROLOGY	Outgoing	OUT OF NETWORK	Y	NULL	External: Pt to Schedule	
9...)	Highland Family Medicine	Family Medicine	OTOLARYNGOLOGY	Internal	NULL	N	NULL	No patient response	
9...)	URMC Orthopaedics and Rehab at Clinton Crossi...	Orthopedic Surgery	PHYSICAL / OCCUPATIONAL THERA...	Incoming	NULL	N	NULL	Expired-Auto Closed	
9...)	Medical Associates of Penfield	Family Medicine	GASTROENTEROLOGY	Internal	NULL	N	NULL	System Automatically Closed	
9...)	Department of Orthopaedics	Orthopedic Surgery	PHYSICAL / OCCUPATIONAL THERA...	Internal	UR	N	NULL	No patient response	
9...)	Medical Associates of Henrietta	Family Medicine	PHYSICAL / OCCUPATIONAL THERA...	Internal	UR	N	NULL	Expired-Auto Closed	
9...)	Penfield Family Medicine	Family Medicine	ORTHOPEDIC SURGERY	Internal	URMFG	N	NULL	All Visits Complete	
9...)	Culver Medical Group	Family Medicine	PHYSICAL / OCCUPATIONAL THERA...	Internal	NULL	N	NULL	No patient response	
9...)	Endocrine Practice Group	Endocrinology	ENDOCRINOLOGY	Internal	NULL	N	NULL	Appointment Scheduled	
9...)	Department of Orthopaedics and Rehabilitation	Orthopedic Surgery	PHYSICAL / OCCUPATIONAL THERA...	Internal	UR	N	NULL	No patient response	
9...)	Calkins Creek Family Medicine	Family Medicine	DERMATOLOGY	Internal	NULL	N	NULL	All Visits Complete	
9...)	Highland Family Medicine	Family Medicine	SLEEP MEDICINE	Internal	NULL	N	NULL	All Visits Complete	
9...)	Highland Family Medicine	Family Medicine	GASTROENTEROLOGY	Internal	NULL	N	NULL	Appointment Scheduled	
9...)	UR Medicine Primary Care - Batavia	Internal Medicine	NEUROLOGY	Internal	URMFG	N	NULL	All Visits Complete	
9...)	LeRoy Medical Associates	Family Medicine	DERMATOLOGY	Internal	NULL	N	NULL	All Visits Complete	
9...)	Highland Family Medicine	Family Medicine	GASTROENTEROLOGY	Internal	NULL	N	NULL	System Automatically Closed	
9...)	Eastside Internal Medicine	Internal Medicine	ORTHOPEDIC SURGERY	Internal	URMFG	N	NULL	All Visits Complete	
9...)	Department of Neurology - Movement Center	Neurology	SLEEP MEDICINE	Internal	NULL	N	NULL	All Visits Complete	
9...)	Calkins Creek Family Medicine	Family Medicine	ALLERGY/IMMUNOLOGY/RHEUMAT...	Internal	NULL	N	NULL	All Visits Complete	
9...)	Physical Medicine & Rehabilitation	Physical Medicine and Rehabilita...	PHYSICAL / OCCUPATIONAL THERA...	Outgoing	Out of Network	Y	Patient Preference	External: Pt to Schedule	
9...)	Endocrine Practice Group	Endocrinology	HEALTHY LIVING	Internal	NULL	N	NULL	No patient response	
9...)	Panorama Internal Medicine	Internal Medicine	GASTROENTEROLOGY	Internal	URMFG	N	NULL	System Automatically Closed	
9...)	Southview Internal Medicine	Internal Medicine	ORTHOPEDIC SURGERY	Internal	NULL	N	NULL	All Visits Complete	
9...)	UR Medicine Primary Care - Hornell	Family Medicine	PHYSICAL / OCCUPATIONAL THERA...	Internal	UR	N	NULL	All Visits Complete	
9...)	UR Medicine Primary Care - Hornell	Family Medicine	GASTROENTEROLOGY	Outgoing	OUT OF NETWORK	Y	NULL	Expired-Auto Closed	
9...)	Medical Associates of Henrietta	Family Medicine	COLON & RECTAL SURGERY	Outgoing	OUT OF NETWORK	Y	Patient Preference	External: Pt to Schedule	
9...)	Olsan Medical Group	Internal Medicine	ENDOCRINOLOGY	Internal	NULL	N	NULL	Appointment Scheduled	
9...)	Genesee Valley Family Medicine - Mt Morris	Family Medicine	SLEEP MEDICINE	Outgoing	Out of Network	Y	Patient Preference	External: Pt to Schedule	
9...)	URMC Speech Pathology	Speech Therapy	SPEECH THERAPY	Internal	UR	N	NULL	All Visits Complete	
9...)	UR Medicine Primary Care - Spencerport	Internal Medicine	DERMATOLOGY	Internal	URMFG	N	NULL	All Visits Complete	
9...)	Calkins Creek Family Medicine	Family Medicine	OTOLARYNGOLOGY	Internal	NULL	N	NULL	No patient response	
9...)	Calkins Creek Family Medicine	Family Medicine	OBSTETRICS & GYNECOLOGY	Internal	NULL	N	NULL	Patient Deferred	
9...)	East Ridge Family Medicine	Family Medicine	DERMATOLOGY	Internal	NULL	N	NULL	Patient Deferred	
9...)	Clinton Crossings Internal Medicine	Internal Medicine	ORTHOPEDIC SURGERY	Internal	NULL	N	NULL	No patient response	

Figure 1: Snapshot of the dataset

REFERRING_SPECIALTY	REFERRED_TO_SPECIALTY	DAYS_TO_SCHEDULE
1	9	59
1	9	2
1	9	2
⋮	⋮	⋮
27	10	2
27	10	14
27	10	51

Table 1: Head and tail of the minimal dataset

referring department to a specialty practice. One patient can have more than one referral in the system, resulting in multiple entries.

The database table, at a minimum, has columns that correspond to the following information: a unique referral ID number, entry data, referring specialty, specialty being referred to, whether or not the referral is internal or outgoing, network status, whether or not the referral was leaked, a reason for referring externally, a reason for closing the referral, and the referral date. There may be more features, but this description is accurate according to our understanding of the original database.

Eventually, it was decided that we will instead be granted access to a more stripped-down and obstructed version of the dataset. This smaller dataset will contain only the essential information– and since the project involves health data, identifying information will be redacted where necessary. This dataset was provided in the form of XLSX and CSV files, directly sent to our team by Dr. Zelenka. Ultimately, if it is decided that we will work with unobfuscated data in the future, we will be provided the data either securely via BlueHive or by remotely connecting to a desktop computer in Dr. Zelenka's office.

The minimal dataset presently includes three columns: *referring_specialty*, which denotes a unique ID number assigned to the referring department; *referred_to_specialty*, which denotes a unique ID number for the specialty being referred to; and *days_to_schedule*, which contains information about the delay in scheduling the referral. Note that the observations in the dataset refer to an individual referral, and thus, duplicate rows can be present. The first three rows and the last three rows of an example dataset are shown in Table 1 to give an idea of what the data looks like.

3 Methods

3.1 Implementation

We used, and plan to continue using, Python for the entirety of the project; that said, if necessary in the future (since the data was originally stored in a SQL database), we are prepared to use SQL queries to transfer data from a database into a format that can be more easily read using Python. The synthetic dataset that has been provided to us has helped us get familiar with the structure of the specialty referral network, while Dr. Fear and Dr. Zelenka are able to use our code on the unredacted data in the meantime. This synthetic data has been provided in an Excel/CSV format, which we then loaded into Python for processing. Eventually, if we are able to work with the original dataset, a secure remote account will be created for our team, providing direct access to the data source. Ultimately, this will be more convenient for the transfer and load operations of data into a Python environment for further analysis.

Throughout the course of the project, we aimed to rigidly follow these steps: **i)** data loading and cleaning; **ii)** performing exploratory data analysis to identify important metrics such as nodes, weights, edges, etc.; **iii)** restructuring/rebuilding the data into a form that is suitable for network analysis; **iv)** build the network visualization as per key metric identified in step **ii**; **v)** analyze the graph with metrics such as node degree, density, centrality measures, etc. to make inferences about the network, **vi)** make necessary adjustments to the network based on the inferred information in step **v** such as anomaly/community detection (if appropriate).

i) Data loading & data cleaning: To load the data into Python, and for subsequent data cleaning and analysis, we plan to use the popular data science library `pandas`. While the extent of potential data cleaning is currently unclear, we anticipate that we will be making great use of this library regardless—for example, it will likely be useful to use the `groupby` method of dataframes to group referrals by department and perform aggregrations on those groups.

ii) Exploratory data analysis (EDA): This will be a critical step as it will help us understand how to relate the nodes in the networks, how frequently each node is appearing, what factors can be important when determining edge weights, node colors, and if any additional information that can be included in the network for further inference.

iii) Restructuring/rebuilding the data: Once we have determined the weights of the graph, we need to restructure the dataframe to accommodate graph design. Steps such as building an adjacency matrix will be used to prepare the data for network design.

iv) Building the network graph visualization: Our strategy for creating the visualization involves utilizing the `networkx` package, which is a popular tool for analyzing network data. The first step is to construct a `networkx` graph, which provides a variety of measurable metrics about the specialty referral network. The graph's layout, such as spectral, random, circular, etc., will need to be identified to optimize node positioning and improve visual appeal. We will also investigate open-source packages such as `pyviz`/`plotly` which can be integrated with `networkx` graphs to produce a user-friendly visualization that enhances the overall user experience.

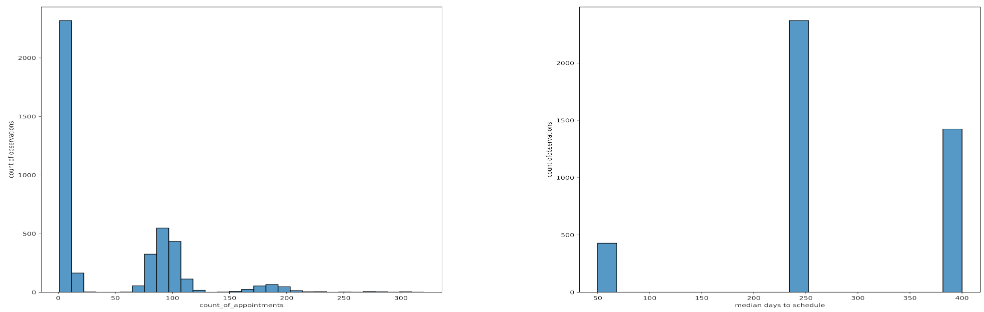
v) Network graph analysis: After constructing the graph network, we will conduct in-depth statistical analysis to identify bottlenecks in the referral specialty network. To accomplish this, we will examine metrics such as nodes, density, clustering coefficient of nodes, and betweenness centrality to gain vital insights into the network's functioning. Our objective is to discover which departments generate the highest number of referrals, which departments are frequently visited in referral chains, which departments take the longest to schedule referrals, and which departments are most likely to fail to complete referrals. We will cross-reference this data with the network visualization created in step **v)** to ensure that the design is consistent and accurate.

vi) Make necessary adjustments: Lastly, we will examine the network visualization to determine whether any modifications can be made to enhance its effectiveness. In our future research, we will explore the potential benefits of implementing algorithms like anomaly detection or community detection to uncover further insights about the referral network's characteristics. We will continue to study the referral network and offer suggestions to improve its performance to the network owners.

Taking into consideration that this is an ongoing research project, we are solidly in the ”network graph analysis” and ”make necessary adjustments” stages. As we are still in communication with Dr. Fear and Dr. Zelenka, we are continuously making improvements and adding functionality to meet their expectations.

3.2 Visualizing the data

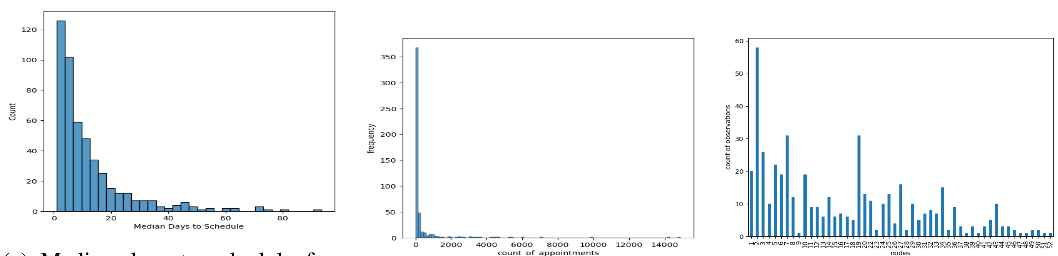
Visualizing the network data quickly becomes challenging as the number of nodes/connections within the network increases. The synthetic network data that represents interactions between various medical center departments had around 150 total nodes with around 200,000 connections among them. Creating a graph with such a huge amount of interactions tends to become quite messy with little to no information provided for the stakeholders looking at the network graph. Thus, it was required to identify key components within the dataset which could inform how to plan the aesthetics of the graph with details such as node size, edges/connectivity importance, color of nodes/edges, etc. This design approach can then provide a more comprehensive network graph to the end users trying to find valuable insights by analyzing the visual at a glance. In the dataset being analyzed, it was noticed that other than the *edge list* (the edge list holds the details that explain how different nodes are connected with edges) details, there was also information such as number of days needed to schedule a referral. While performing EDA on the dataset, it was possible to uncover information



(a) Referral count distribution

(b) Median days to schedule distribution

Figure 2: Histograms of referrals and median Days to Schedule



(a) Median days to schedule frequencies

(b) Referral count frequency

(c) Node frequencies

Figure 3: Frequencies of variables that influence aesthetics

such as the number of repetitive referrals being made from the same departments, details of the departments only creating referrals compared to the ones only receiving referrals, and the number of days usually required to schedule a referral appointment along with its variance across departments. These details were then found to have the potential to pivot the raw network information into a more compact/digestible format with a cleaner and more concise visual.

Through exploratory data analysis, we found that there were around 100 nodes which only made referrals to other departments (out-degree), 76 nodes which were only at the receiving end of referral (in-degree), and 26 nodes which had both out and in-degree details. Thus, it was decided to color these nodes separately. The count of referrals made/received from each department was calculated and used as the edge weight indicator. Because the count of appointments hugely varied across departments, this information was grouped into a bin of four, i.e., 50, 150, 250, and anything above 250 (see Figure 2a). The bin sizes/counts were determined by analyzing multiple histograms of different bin sizes and then finalizing the one which captures the most variance with the least number of bin counts. Finally, the highest weights were given to the group with highest number of referrals, whereas other groups were given subsequently lower weights.

Departments were involved in multiple referrals, therefore, for each sending/receiving department, the median number of days to schedule was calculated and grouped into three bins with bin sizes of 50, 250, and 250+ days due to its high variance (see Figure 2b). Bin sizes/counts were determined by analyzing multiple histograms of different bin sizes and then choosing the one that captures the most variance with least number of bin counts. This information was used to represent edge color, where edges of one color correspond to the 50-bin, edges of another color correspond to the 250-bin, and edges of a third color correspond to the over-250-bin.

To summarize the exploratory data analysis and how it affected the visualization, there are three main aesthetic choices that were made in order to provide a more intuitive and meaningful experience for the user. Firstly, we decided that a node's color would be determined by the direction that referrals were flowing in relation to that node; i.e., node that only make referrals, nodes that only receive referrals, and nodes that do both would be colored differently. Secondly, the size of a node would be proportional to the number of connections that are associated with that node, and it would take the in and out-degrees of each node into account as well. Lastly, the color of an edge would be decided by the median number of days it takes to schedule a referral between two departments. Exact numbers, bins, and boundaries for each of these choices were decided by using the plots shown in Figure 3.

While experimenting with different graph layouts, it was found that the *spring* layout gave the most comprehensive results as it assigned the nodes according to their size in the graph space, placing smaller nodes towards the outside and larger nodes towards the inside. The Plotly python library was utilized to feed all of the parameters discussed above to create a much more comprehensive graph for our stakeholders. An additional feature was added which allowed users to hover over edges to get information at a glance regarding in/out-degree, median/min/max days to schedule, number of appointments, etc., which our stakeholders found to be very helpful. To summarize, at a single glance, the network graph is able to provide details such as the most important actors in the networks, the departments with most/least number of referral appointments and the departments taking the most/least time to schedule referrals appointments.

To improve this further, we aimed to make the visualization more granular so that users would be provided the flexibility to analyze interactions within a group of particular departments of their choice without the distraction of irrelevant network data. We also chose the the colors used in the visualization to make it more universally accessible, specifically in regards to those with colorblindness; for example, instead of using red, yellow, and green for node colors, we used brown, yellow, and blue.

3.3 Interactivity via Flask

During our weekly meetings with Dr. Fear and Dr. Zelenka, we discussed the possibility of creating a web application using Flask—a lightweight web framework for Python—to provide an end-to-end solution for analyzing the speciality referral network. The web application, in combination with the innate interactive of a Plotly visualization, will allow laypeople with no prior coding experience to change datasets, create the visualization, and interact with it as they see fit.

The Flask app, at a minimum, required the following functionality:

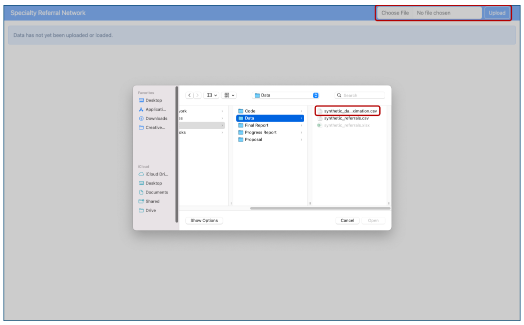
- Upload new datasets: There should be a way to upload a dataset directly, without having to hard-code any specific filepath. This will be accomplished through the use of a simple form on the application, containing only a button to browse to the data on the local file system and a button to upload the data. Once the data is uploaded, this should immediately initiate generation of the visualization.
- Visualization of the network: The visualization should be uncluttered, yet contain enough information to make observations and extract useful information about the specialty referral network. The visualization was explained more in-depth in Section 3.2.
- Clean UI and intuitive UX: The look of the web application user interface should be clean, minimal, and ultimately, aesthetically pleasing. The user experience should be simple and intuitive so that anyone can understand how to use the application. For example, the upload button should be in an obvious, yet still unobtrusive, location, so that it does not detract from the visualization (the focal point of the application).

The current status of the Flask application is described in more detail in the following sections.

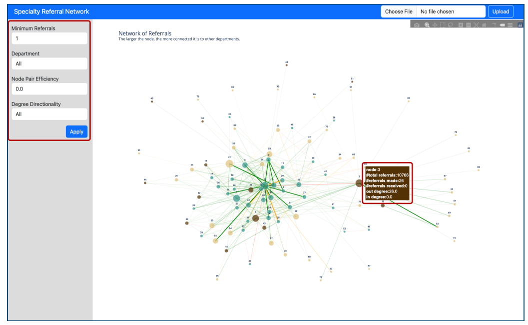
3.3.1 Uploading a file

While having the visualization itself was sufficient in allowing us to make inferences about the data, the user experience was certainly able to be improved. Every time the data file changes, the data needs to be replaced on disk, a code editor or IDE needs to be loaded, and the visualization needs to be regenerated, among other things, which is something a layperson might be able to learn given enough time and repetitions, but might not actually be feasible in practice.

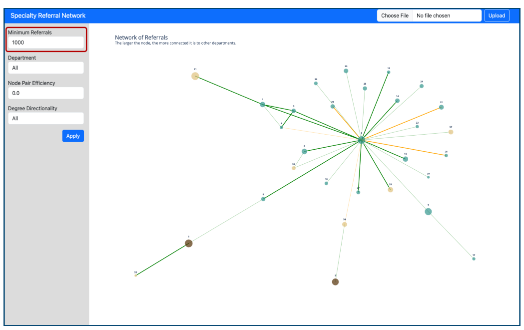
The main benefit of having the visualization live in a web application is that it's easy to use for a layperson—nearly everyone knows how to work a website—but it also allows for plenty of addi-



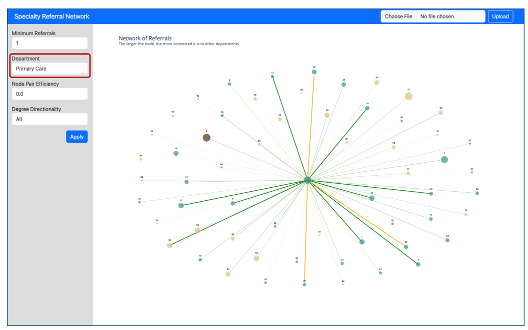
(a) Uploading data



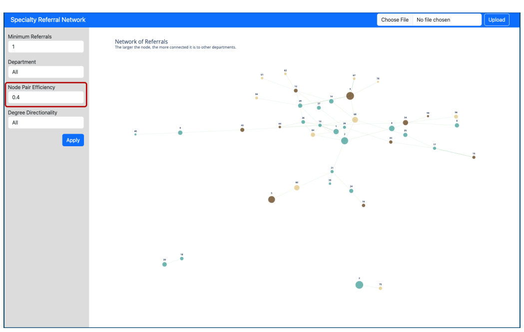
(b) Layout of the application



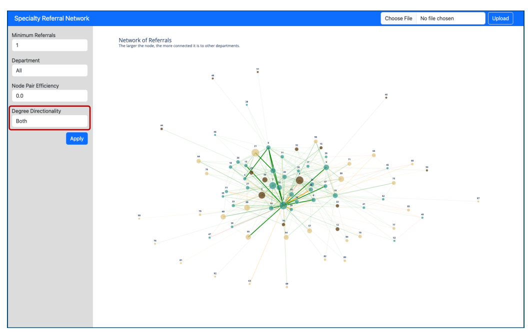
(c) Minimum referrals filter



(d) Department filter



(e) Efficiency filter



(f) Degree directionality filter

Figure 4: Flask application functionality

tional functionality, namely the ability to easily change out data. As shown in Figure 4a, this can be done by using the upload button on the application.

Another benefit is that, once the data has been uploaded and the visualization has been generated, the visualization is stored on disk. This means that you can close and reopen the application and the visualization will load immediately without any noticeable overhead.

3.3.2 Layout of the application

One of our main goals with the Flask application was to make the visualization more interactive and customizable, so that the user can focus on certain parts of the network without being distracted by an abundance of irrelevant information. We accomplished this through the use of adding a sidebar, shown in Figure 4b, which includes various filters and thresholds.

It's also important to note that this is in addition to the interactivity and functionality that's built-in to Plotly, the library that the visualization was built with. By default, Plotly allows us to pan the visualization, zoom in and zoom out, and many other things. It also provided the opportunity to add tooltips to nodes and edges in order to provide more information, such as the in and out-degree of nodes.

3.3.3 Filtering by number of referrals

The first feature that was made possible by the Flask application and included in the sidebar was the ability to set a minimum number of referrals for a connection to be drawn, shown in Figure 4c. Obviously, this can be done by changing the code for the visualization itself, but the web application allows the layperson to change it quickly and easily, without interacting with the backend at all.

This feature in particular is helpful because it allows the user to essentially remove noise or, in other words, connections that are deemed unimportant.

3.3.4 Filtering by department

The user is also able to filter the visualization by department as shown in Figure 4d. Essentially, choosing a department from the drop-down menu would reduce the scope of the visualization and only consider referrals that are made or received by that department.

3.3.5 Filtering by efficiency

As seen in Figure 4e, it's also possible to filter by efficiency, which is a metric we defined as the number of referrals exchanged between two departments divided by the total number of days it took to schedule those referrals. After we performed that calculation for each node pairing, those efficiencies were then min-max normalized so that the minimum value was 0 and the maximum value was 1.

This filter is especially useful when trying to determine which departments regularly process referrals more efficiently, which is a major step forward when trying to identify where bottlenecks are occurring.

3.3.6 Filtering by degree directionality

The last filter that we currently have implemented, which we've called degree directionality, is essentially a way to visualize which departments only make referrals and which ones only receive referrals. This could prove useful when trying to determine, for example, if a bottleneck is caused by a department that makes a lot of referrals, or by a department that receives a lot of referrals. This filter is highlighted in red in Figure 4f.

3.4 Evaluation

Dr. Fear and Dr. Zelenka have not requested any specific, measurable metric of success for this project, as the goals for this project were initially quite unclear and were subsequently formed as the project evolved. However, they have agreed that the visualization should be relatively simple and easy to understand. If the visualization can be shown to a layperson with no clear familiarity with the subject matter and still be understood, and if the Flask application can be used without having to interact with the backend of the Flask application, then the project will be considered a success.

4 Conclusion

We've learned through regular, constructive feedback from Dr. Fear and Dr. Zelenka that the visualization we have created will be beneficial to UR Medicine in the long term, and that the Flask application is a suitable medium for interacting with the visualization. It has also become clear that the additional accessibility and customizability that was made possible by the Flask application is advantageous when attempting to identify where bottlenecks are occurring in the speciality referral network. Overall, our work on this project has laid a promising foundation for further network analysis and research.

In the future, there is a slew of improvements we could make to both the base visualization and the high-level Flask application. For example, the sidebar is, at the moment, a highly-functional proof of concept that could be expanded upon greatly with more features and functionality; one such feature that could be particularly useful is the ability to somehow identify communities within the network and filter out irrelevant communities. We could also securely host the Flask application on UR servers and assign it a domain, so that it is much more easily accessible by the team members at UR medicine while remaining unavailable off-site. Just to reiterate, this project is ongoing, and is thus an ever-evolving effort to provide the best results and user experience.

Acknowledgments

We thank Dr. Kathleen Fear (UR Medicine) and Dr. Justin Zelenka (UR Medicine) for providing the opportunity to conduct this research, including useful discussions and providing data.