On Detection of Influential Users and Content on Reddit*

Rollen S. D'Souza[1]

***Abstract*— Fill in abstract.Fill in abstract.**

I. INTRODUCTION

Talk about goals of project, pathway and where we ended up.

II. BACKGROUND

Talk about graph theory, centrality and Laplacians.

III. MODELLING

A. Assumptions

In order to reason about influential content without temporal information or content analysis, we make the following simplifying assumptions:

Assumption III.1. Users are influenced (positively or negatively) by content they read.

Assumption III.2. Users comment only on content they have read.

Assumption III.3. Content is influenced by the content creator.

Assumption III.4. Deleted users and content are not influential and do not affect other users in a meaningful way.

Assumptions III.1 and III.2 are the most contentious. They effectively assert that users engage with the platform in an intellectually curious manner. This is, at face value, absurd. On the other hand, the case can be made that even if users don't fully read content that they interact with, they at, the very minimum, skim and grasp the general gist of the content. This relaxation of the assumption is sufficient but doesn't lose the essential structure of influence implied by the assumptions. Specifically these two assumptions imply that we can infer a user is influenced by content when we detect a comment made by the user. As a result, we take the first two assumptions as reasonable. Assumption III.3 is easy to see as reasonable. Even if the content posted is not owned by the user — that is, a link to some other content — the user is still introducing the content to the community and, as a result, acts as the influencing agent. It is not apparent at first glance that Assumption III.4 is reasonable. We discuss this later in the work and will establish its validity.

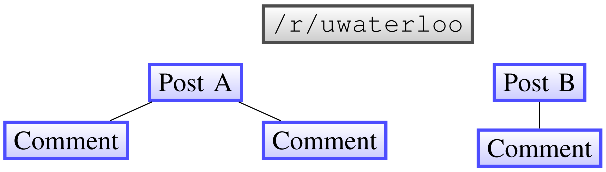

Fig. 1. Graph-like structure of the Reddit platform.

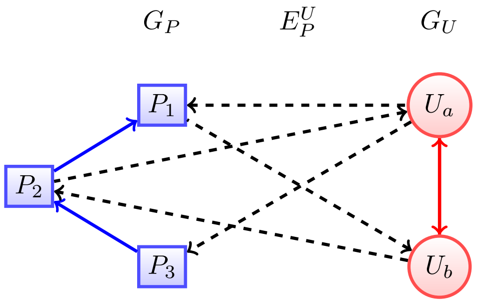

Fig. 2. Graph structure showing the relationship between users and submissions/comments. G_P are those nodes and edges that are blue. G_U are those nodes and edges that are red. E_P^U is the set of edges denoted by broken lines.

B. The Structure of Reddit

Reddit is a platform for users to submit content, named submissions, and to have other users comment on them. These submission come equipped with a title and either a body text, media or link to external content. These submissions are organized into large communities, known as subreddits, that which users may subscribe to. The structure can be visualized as in Figure 1. Explain other possible related structure. Upvotes and Downvotes. Controversial

C. Graph Model

Two models are used for analysis. The first models how influential content or users are on a given subreddit as a connection between two graphs. A visualization of the structure can be found in Figure 2. Graph G_P is the graph that captures the inherent structural relationship between submissions/comments. Assumptions III.1 and III.2 imply that a given submission is influenced by its comments. More generally, any parent content is directly influenced by its children. These are visualized as the blue edges in Figure 2. This is augmented using edges that relate users to submissions/comments. The second model attempts to capture which users converse with each other. This graph, G_U, can be thought of as being generated by the graph created above.

*This work was not supported by any organization

[1] Rollen S. D'Souza is a graduate student in the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Waterloo, rollen.dsouza@uwaterloo.ca

1) Graph Model for Centrality Analysis: Denote the set of users by U and the set of submissions and comments by P. We first define the existence of the digraph $G_P = (P, E_P)$ that denotes the relationship between submissions and comments. E_P is constructed in the manner described in Algorithm III.1. We give edges in E_P a universal weight of 1. Weighting using upvotes was considered but was determined to yield results that were not desireable. For one, it was not clear that upvotes were a meaningful method of ranking influence. Further complexity arises when considering controversial content, which has a mix of upvotes and downvotes. A highly controversial topic may be highly influential, which contradicts the idea that upvotes correlate with influence. Possibly given knowledge of the number of voters this problem could be alleviated. However, since not all subreddits release data about how many users voted, it is simpler to reduce the complexity by eliminating any non-trivial weighting method. Observe that G_P is a forest where the submissions act as roots of the trees.

Algorithm III.1: Constructing E_P.

```
forall parent $\in P$ do
    for child $\in P \cap ChildrenOf(P)$ do
        $E_P \leftarrow E_P \cup \{(\texttt{parent}, \texttt{child})\}$;
```

This structure is not interesting by itself and we need to extend it in order to facilitate analysis. The set of edges E_P^U, that connects submissions/comments to users, is what brings the significant cross-post structure. E_P^U is generated in the manner described by Algorithm III.2. The rules are that: a comment author is influenced by the submission (root in G_P) and a comment author influences the comment. A more convoluted structure may be derived by instead having the comment author be influenced by the immediate parent comment. This structure was not investigated and could yield differing results. The argument for choosing the structure defined here is that authors may bring the whole context of the submission to a given comment, and not just the parent comment. The directed edge from the author to the submission allows us to model this sort of behaviour. That is, the user is influenced by all content on the submission page. This does ignore temporal effects, and so could yield incorrect results. On the other hand, the model not investigated is not as susceptible to temporal effects as it directly captures the temporal structure. However, it leaves out the ability to model any sort of larger contextual awareness the user may have. The structure described above may be augmented with self-edges for all users and aperiodicity of the resulting graph can be guaranteed.

Algorithm III.2: Constructing E_P^U.

```
forall user $\in U$ do
    for $p \in P$ and user $= AuthorOf(p)$ do
        root $\leftarrow RootOf(p)$;
        $E_P^U \leftarrow E_P^U \cup \{(p, \texttt{user}), (\texttt{user}, \texttt{root})\}$;
```

Claim III.1. Let the set of vertices be $N = P \cup U$. We also let $E_N = E_P \cup E_P^U \cup \hat{E}$ be a set of edges, where $\hat{E}$ is the set of directed self-edges for all users in U. Then $G' = (N, E_N)$ is an aperiodic graph.

Proof. Easy to see by the fact that cycles of any length may be constructed by including the users in the cycles. The simplest cycle may be constructed (see Figure 2) using a commenting user and the associated submission. □

2) Graph Model for Cluster Analysis: Submissions and comments are not directly significant for identifying subgroups within a community of users. The model used here is restricted to a weighted digraph $G_U = (U, E_U, W_U)$. However, we desire a method of constructing relationships between users that capture "interaction" between users. Algorithm III.3 defines one such construction. It captures the idea that a user Bob is influenced strongly by another user Jenn if Bob comments a lot on content Jenn submits. The algorithm adds an additional requirement: that the interactions are unique modulo the root submission. This additional restrition attempts to count the number of "conversations" taking place between users without the need for explicit start and stop points. Complete explanation of the graph used for cluster analysis.

Algorithm III.3: Constructing E_U.

```
forall $p \in RootsOf(P)$ do
    conversations $\leftarrow \emptyset$;
    for $p_2 \in ChildrenOf(p)$ do
        ;
```

IV. Implementation

There are two methods of collecting Reddit submission and comment data. There is a live application programming interface (API) available for use, that provides live up-to-date data. The most natural API request available for downloading submissions is limited to the last 1000 submissions on a given subreddit. This substantially limits the amount of data available for collection; for some subreddits this constitutes only a few weeks of data. Other methods to getting this old data exist, but involve using the API in a convoluted manner. A user find reddit user addressed this problem and collated all data on Reddit for over ten years for the purpose of data analytics. The data can be found online, freely available for use, in an archived format [1]. Submissions and comments are stored in separate archives, organized by months or days. The data is stored in these files as compressed, line-delimited Javascript Objects (JSON).

The analysis focusses on data for the time frame starting at November 2017 and ending in February 2018. I developed a Python script that filters out unnecessary data from these archives and inserts the useful data into a Redis database, a fast in-memory key-value database [2]. The software developed for this project and used for analysis is publicly available on Github [3].

A. *Centrality Analysis*

For measuring the influence of any given user or content, I propose using the Katz centrality measure. Centrality measures provide a measure of how "central" a node in a graph is. The simplest of measures simply counts the in-degree of a node. The Katz centrality measure is defined as,

Definition IV.1. Let $A \in \mathbb{R}^{n \times n}$ be an adjacency matrix for an aperiodic graph G. Let $0 < \alpha < \rho(A)$. The Katz centrality measure for node i is defined as,

$$c_i = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k \left(A^k\right)_{j,i}.$$

This measure counts the number of paths from all nodes in the graph to a given node i, scoring paths of shorter length with a higher weight. The graph G' was shown in Claim [**?**] to be aperiodic and so we can use the Katz centrality measure on the induced adjacency matrix. Since G' is unweighted, we choose the natural weighting of 1 for all edges in the adjacency matrix. The implementation used is a custom script that employs an iterative technique that converges to the Katz centrality measure for all nodes [4].

B. *Cluster Analysis*

Spectral clustering is a family of techniques used for identifying disconnected components in a graph or, more generally, clusters of highly connected nodes using the graph Laplacian matrix. A survey of these techniques can be found in [5]. I instead follow a variation of the simplest technique described in [6], [7]. We recall significant results used for spectral clustering now,

Definition IV.2. Let $A \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{n \times n}$ be the adjacency and out-degree matrix for the graph G_U. The graph Laplacian is defined as,

$$L = D - A.$$

The normalized graph Laplacian is defined as,

$$L_{rw} = I - D^{-1}A.$$

Theorem IV.1 (Properties of the Graph Laplacian [4])**.** The graph Laplacian and the normalized graph Laplacian enjoy the following properties,

(a) Say graph G_U has n nodes with k connected components. Then

$$\operatorname{rank} L_{rw} = \operatorname{rank} L = n - k.$$

(b) The vector of all ones, $\mathbf{1} \in \mathbb{R}^n$, is in $\ker L_{rw} = \ker L$. This vector evalutes the membership of all nodes to the graph.

(c) The eigenvalues λ_i of L and L_{rw} can be arranged in the following manner,

$$0 = \Re\{\lambda_1\} \leq \Re\{\lambda_2\} \leq \cdots \leq \Re\{\lambda_n\}.$$

The methodology to find a cluster within a graph is to bootstrap a clustering technique used in $\mathbb{R}^m$, the k-means algorithm, by the eigenvectors of L_{rw}. This relies on the observation that the eigenvectors associated with eigenvalues at 0 are those that denote membership to a given connected component of the graph. Moreover, eigenvectors near zero can be thought of as measuring a partial membership to a cluster. As a result, we wish to find the eigenvectors of L_{rw} whose eigenvalues are near 0. This is not the best approach due to numerical instability of the eigenvalue algorithms used for sparse matrices. The choice to use L_{rw} alleviates the problems introduced by having a poor condition number, however we need to also employ a technique discussed by Frederix and Barel. They suggest instead looking at the matrix [8],

$$\hat{L}_{rw} = D^{-1}A.$$

The matrix $\hat{L}_{rw}$ has the same eigenvectors as L_{rw} while the eigenvalues shift. The eigenvalues instead move to be less than, or equal to, 1. This reformulation turns out to be much better suited to the numerical algorithms implemented to find eigenvalues of large sparse matrices.

V. ANALYSIS

One subreddit of particular interest to myself is the `/r/uwaterloo` subreddit. This community consists to date over 26 thousand subscribers. The subreddit is famous for a goose-related subculture that resulted in a man receiving a goose tattoo on an unsavoury location of his body [9].

ACKNOWLEDGEMENT

The author thanks Professor John Simpson for offering the special topics course on Networked and Distributed control, thereby providing the essential background for this analysis. The author also gives a special thanks to Jason Baumgartner for uploading archived Reddit API data to [1].

REFERENCES

[1] J. Baumgartner, "Directory contents." [Online]. Available: https://files.pushshift.io/reddit/

[2] "Redis," July 2017. [Online]. Available: https://redis.io/

[3] R. D'Souza, "rollends/red-prism." [Online]. Available: https://github.com/rollends/red-prism

[4] F. Bullo, *Lectures on Network Systems.*, 1st ed. CreateSpace, 2018. [Online]. Available: http://motion.me.ucsb.edu/book-lns

[5] M. C. Nascimento and A. C. de Carvalho, "Spectral methods for graph clustering – a survey," *European Journal of Operational Research*, pp. 221–231, 2011. [Online]. Available: doi:10.1016/j.ejor.2010.08.012

[6] R. Horaud, "A short tutorial on graph laplacians, laplacian embedding, and spectral clustering." [Online]. Available: https://csustan.csustan.edu/ tom/Clustering/GraphLaplacian-tutorial.pdf

[7] U. von Luxburg, "A tutorial on spectral clustering." *Statistics and Computing*, vol. 17, no. 4, 2007.

[8] K. Frederix and M. V. Barel, "Sparse spectral clustering method based on the incomplete cholesky decomposition." *Journal of Computational and Applied Mathematics*, Jan. 2012. [Online]. Available: doi:10.1016/j.cam.2012.07.019

[9] `/u/SsseCn8jx`. Fuck it, if this post receives 2000 upvotes i will tattoo a goose on my ass. [Online]. Available: https://www.reddit.com/r/uwaterloo/comments/5ibr95