Analyzing conferences in Twitter with Social Aviary

Ryan Noon and Hamilton Ulmer
Stanford University
CS 322 - Network Analysis

ABSTRACT

Utilizing the Social Aviary, we analyze the network implied by usage of the conference hashtag `#online09` on Twitter. We observe that, if taken as a network, the group of users who tweet with the requisite hashtag over the course of a conference does not get particularly more dense, though it does densify relative to the immediate users connected to the conference-goers who did not use the hashtag. This is the start of a much large project, analyzing how these types of events create a cover over a subset of communities and then densify over that coverage.

1. INTRODUCTION

We are approaching a golden age in the analysis of networks. Online services like Facebook, LinkedIn, and Twitter provide us with a varied and rich data on the structure and dynamics of interactions, some of which have some meaningful bearing on real-world networks. Researchers have found ways of simulating and analyzing the network topologies of various types of real-world networks, to massives scales. Understanding the dynamics of real social networks over time is, however, still a challenge. Many studies compare two snapshots of the same network, and from that devise metrics to understand the evolution of the network, as noted by [7].

We take a different approach. Using the Twitter API we built a toolset for analyzing the networks implied by the tweets of conference participants. Because conferences, especially social media conferences, strongly suggest to their participants that they tweet about the conference with a hashtag, we can find who is at a conference, who they know, and who they follow. We have the exact timing of edge and vertex creation of interesting subnetworks that develop very quickly.[1] We discuss more of our assumptions and provide a brief description of how conference participants use hashtags in section 3. Because we think the toolset we built could be a very useful network data collection and analysis tool for communities on Twitter in general, we hope to refine it more and release it as an open source project. We discuss our implementation, called Social Aviary, in [10].

Recording and analyzing conferences through Twitter has several advantages. First, while it is very difficult to track the entire evolution of the Twitter graph, collecting data on connections formed at a conference is relatively straightforward, by comparison. The Twitter API, described in 3.2, is sophisticated enough to be utilized to collect data on conferences. Second, if we restrict our set of viable conferences to *social media* conferences (i.e. those that focus on understanding the impact of social network sites and services), we can not only more accurately approximate the conference network, but also capitalize on the relative abundance of social media conferences that occur weekly.

2. RELATED WORK

Some work has been done on empirically analyzing networks over time at such fine detail. The study most similar to ours, Kossinets and Watts [5], analyzes a registry of emails sent by 43,553 undergraduates and graduates at a major university over the course of one school year. The main analytical quantities of interest appear to be the empirical probability of triadic closure (defined as a cycle of length 3) and cyclic closure more generally (defined as the closing of a cycle of length d by adding one edge). The authors find that micro-instability in portions of the network "average out" and create some larger, emergent macro-trends.

Our work relates to the study of community structure in large networks by Leskovec et al. [8]. A few points are worth mentioning. First, they estimate that communities of less than around 100 vertices tend to exhibit much greater exclusivity relative to the rest of the network. Past 100 vertices, the community tends to connect more with other vertices outside of it. In our case, we assume that vertices in the conference hashtag network are already part of some community. The conference should theoretically never be the sole community in which vertices belong. We can, nevertheless, examine the clustering behavior within these conference hashtag networks.

3. WHY TWITTER MAKES AN ATTRACTIVE RESEARCH PLATFORM

[1] Our back-of-the-envelope estimate suggests that there are as many as 15 conferences a month that strongly suggest to participants to use a conference hashtag.

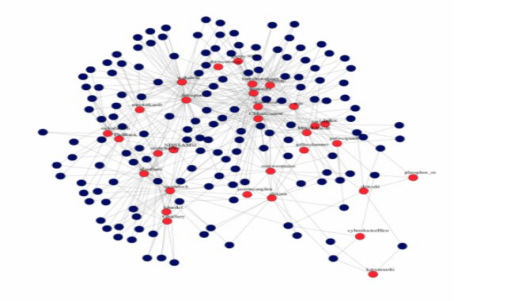


Figure 1: The network formed from the messages and relationships among R programmers on Twitter.

3.1 The Emergence of Niche Networks

Because of its unstructured, broadcast-centric nature [4], people can use and abuse Twitter to many ends. One of the major use cases has become the natural formation of *niche networks*, small communities of users that share a similar interest, be it for leisure or professionally. Twitter's lean nature makes the dissemination of information and discussion through these niche networks valuable for users.

3.1 illustrates a niche network on Twitter. The data comes directly from the Twitter API, pulled on search terms. This network represents users of the programming language and environment `R` on Twitter. This is a two-core representation of the network. In red are the `R` users, and in blue are the users who they all mutually follow. [2]

3.2 A Powerful API

From a data collection perspective Twitter has a powerful Application Programming Interface (API). Twitter has since its inception been keen on opening up its data streams to researchers and companies. For business this has led to an explosion of Twitter-based applications and clients. Researchers have traditionally been successful in obtaining portions of social networks from companies that run them for analytical purposes.

Yet there are practical issues involved in asking for a static data set. We might not necessarily get the date and time of a new follow, for instance. This naturally makes it difficult to record and analyze the growth of a new community within the social graph.

Luckily these practical issues can be resolved by utilizing the same API that companies have been using for years. While a researcher cannot measure the dynamics of the entire social network through the API, one can answer specific questions about the growth of certain communities.

3.3 Hashtags: community indicators

The final relevant feature, and for our purposes the most important, relates using hashtags in a user's tweets. Hashtags are a way of loosely categorizing a tweet. The convention is very simple; all a user needs to do is include `#tag` in their tweet. In fact, the network in figure Fig. 3.1 came from using the Twitter API to search for instances of `#rstats` in tweets.

Hashtags are important because, for people who care about certain fields, communities, products, concepts, languages, or events, hashtags become unique focal points for categorizing the volume of content generated by users. They can define communities, depending on the purpose of the hashtag. We have given one example of this, through the `#rstats` hashtag.

Conference organizers have capitalized on users' familiarity with hashtags by suggesting to conference attendees they use a specific hashtag so that others, both at the conference and those who cannot attend, can view the *back channel* of information. This has become a powerful use case for hashtags. It makes connecting with others at a conference, as well as keeping up with the most interesting events, much easier. It also helps define a community of people who have shared an experience at a conference.

4. TIME SERIES COMMUNITY ANALYTICS

Because in our case the issue of identifying a potential community has been side-stepped with additional metadata, and because we know the timing of edge creation and node admittance to communities, There are several interesting questions we can answer for a given community.

4.1 Common Notation

We will first define some common notation. We define a graph G, along with its set of vertices, V, and set of edges, E. The number of vertices and edges are $|V|$ and $|E|$, respectively. $v_i \in V$ denotes vertex i, and $e_{ij} \in E$ represents an edge that connects vertex i to vertex j. We also define the adjacency matrix as W, where $w_{ij} \in W$ denotes the strength and presence of an edge between two vertices, assuming $w_{ij} > 0$. Finally, we denote D as a diagonal matrix of degrees, where $d_i \in D$ is the degree of vertex i.

4.2 How do the number of vertices, and the average degree, change over time?

A first set of questions involves typical metrics one might calculate for a static graph. We want to understand how many vertices enter the network, and their timing relative to the real-world events underlying the conferences.

We also look at the average degree over the course of the conference. We hypothesize that the average will greatly increase over the span of the conference. This seems rather intuitive and obvious, so we also will look at how the median degree changes over time.

4.3 How does the density of the network change?

The graph density is defined as

$$d = \frac{2|E|}{|V|(|V|-1)}$$

Where $|E|$ is the number of edges in the graph and $|V|$ is the number of vertices. The density sits between 0 and 1. We would expect the network to become more dense over time.

[2]Image courtesy Drew Conway: `http://www.drewconway.com/zia/?p=1471`

4.4 How "cliquish" is the graph?

To measure the cliquishness of the graph, we measure the *average clustering coefficient*, first proposed by Watts and Strogatz [12]. The authors use the term "cliquish" in describing what a high clustering coefficient could be interpreted. We define it as

$$c_i = \frac{|e_{ik}|}{|N_i|(|N_i| - 1)}$$

Where $N_i = \{k : k \text{ is a neighbor of } i\}$, and k is the index of a vertex v_k.

We are interested in finding the average clustering coefficient,

$$C_G = \frac{1}{|V|} \sum_{i=1}^{n} c_i$$

4.5 Do more edges form within the network or between it and vertices outside of it?

Ideally, as new nodes enter the system and new edges are created within the niche network as well as out of it, the ratio of inter- and intra-cluster density over time should change. Does it grow bigger or smaller?

For each rendered graph we calculate the *Krackhardt's E/I Ratio*, defined as

$$K = \frac{|U| - |E|}{|U| + |E|}$$

here $|U|$ is the number edges in the network pointing to external vertices and $|E|$ is the number of edges that point to other edges within the network [6]. K takes on values between -1 and 1, though the interpretation is clear: if $|U|$ is bigger than $|E|$, we know that outward-connecting edges account for $K\%$ more of the total inner and outer connecting edges. A similar interpretation exists for when $|E|$ is bigger than $|U|$. For the sake of clear exposition and uniformity of common notation we replace the E and I with $|U|$ and $|E|$, respectively.

This measure shares some similarities to the inter- and intra-cluster densities outlined in [2]. Both can give us a decent sense of how isolated a community is, since they and many other functions of inner and outer edges all measure the same thing. We choose the E/I ratio because of it is easy to interpret and because it allows us to focus on the local connections vertices in our network have with vertices outside of it, rather than how locally the conference network is compared relative to the entire Twitter graph.

4.6 How robust is the network over time?

There are a few ways to measure the robustness of a network. Here, we choose to calculate the algebraic connectivity over time. The algebraic connectivity is defined as the second eigenvalue of the normalized graph Laplacian, first proposed by Fiedler for regular graph Laplacians [1].

Recently spectral methods, which depend on the eigendecomposition of the graph Laplacian, have become very popular, most notably with community detection in graphs and clustering methods [11].

4.6.1 Algebraic Connectivity and The Multiplicity of Zero Eigenvalues

Suppose that $\lambda_i, i = 1, ..., n$ are the eigenvalues of a normalized graph Laplacian $L = I - D^{-1/2}WD^{-1/2}$, where W is the weighted adjacency matrix of our graph G, and D is a diagonal matrix of degrees. We assume the eigenvalues have been sorted by size, where $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. The sorted eigenvalues of this matrix tell you a few useful things:

The number of zeroes tells you the number of disconnected parts of the graph there are. If, for instance, $\lambda_1 = \lambda_2 = \lambda_3 = 0$, then we would expect there to be two connected components in the graph.

The size of the first nonzero eigenvalue gives you the *algebraic connectivity* of the graph. When a graph is connected then this is usually the second eigenvalue. In real-world applications, however, there are often isolated nodes and splits in the community. Because our questions concerning the number of isolated nodes gets answered elsewhere, we throw them out before calculating the eigenvalues of the Laplacian.

Its relation to the diameter of the graph There are many bounds, both upper and lower, relating to the algebraic connectivity and the diameter of a graph. A lower bound from [9], for example, is:

$$\text{diam}(G) \geq \frac{4}{n\lambda_2}$$

Similar quantities exist for upper bounds.

Robustness The algebraic connectivity has been shown in simulations to be a decent measure of a network's robustness against random edge deletion [3]. The higher the algebraic connectivity, the more robust the network is to random deletion.

4.6.2 The Eigengap Heuristic and Clusters

We will also use the *eigengap heuristic* to identify the total number of possible clusters in the data. There is to date no theoretical justification for using it, but practically the results tend to work well [11]. The idea is to look at the ordered eigenvalues, $\lambda_1, \lambda_2, ..., \lambda_n$, and pick a small value for k such that

$$k^* = \max_k |\lambda_k - \lambda_{k+1}| = \Delta_{k+1} \text{ subject to sanity constraints}$$

This heuristic method, though having no theoretical justification, can work well in identifying the number of subcommunities in a network. This gives us a sense of how many clusters might exist over time. We will also use the actual value of the eigengap heuristic as a proxy as a rough confidence metric for there being more than one cluster.[3] The smaller the value, the harder it is to discern the number of clusters, likely because the similarity matrix suggests that the vertices are highly similar. This is something implied by von Luxberg, though not implicitly stated [11].

[3]A future paper by the authors on this topic still currently in an early stage.

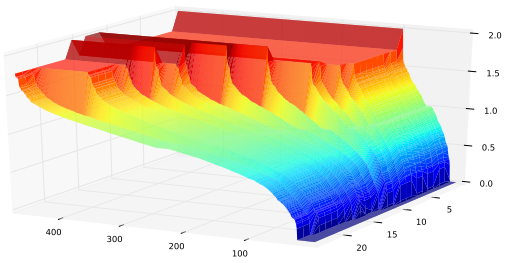


Figure 2: The sorted Laplacian eigenvalues as they grow over time for a conference network

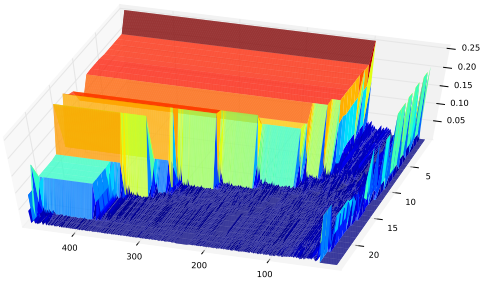


Figure 3: Differences between the sorted eigenvalues of the same Laplacian over time. Notice the large corridor between the peaks.

In practice, we must find a decent decision rule for finding the appropriate eigengap. "Subject to sanity constraints," (paraphrasing the discussion of this method in [11]) works fine if we are in the position of eyeballing the appropriate value of k. Automatic picking does require some algorithm, especially when we are interested in plotting many eigengaps over time. We propose a simple method for finding the eigengap heuristic.[4] First, for each period of our rendered network, we plot the sorted Laplacian eigenvalues for a conference network over its course, as shown in Fig 4.6.2. We notice that, for this real-world network, the slopes of the sorted eigenvalue curves are steepest at the front and at the end. If we look at the differences between successive eigenvalues, $|\lambda_i - \lambda_{i-1}|, i = 2, 3, ..., n$, the relationship we can utilize, depicted in Fig. 4.6.2 becomes clearer. We then take the largest earliest difference Δ_{k+1} (easily done by splitting the differences down the middle of the corridor depicted in Fig. 4.6.2), and pick the associated value k as the number of subcommunities.

This suggests a simple algorithm for finding the appropriate value of k. For each time period calculate the differences in the sorted eigenvalues, cut the vector in half, and find the largest difference from the half associated with the lower eigenvalues, along with its corresponding k. The eigengap heuristic and the algebraic connectivity can give us some sense of the narrative of the graph over time. As the algebraic connectivity grows or shrinks we would expect the eigengap to move in the opposite direction.

[4]In the interest of space we defer discussion of the simulations we ran to determine how robust this method is.

5. A CASE STUDY - ANALYZING CONFERENCES THROUGH Social Aviary

5.1 `#Online09` - The IMS Conference

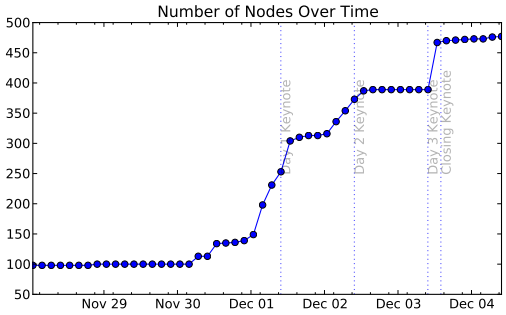
The Online Information 2009 Conference Was held from the 1st to 3rd of December, 2009, in London.[5] Writers, entrepreneurs, and personalities presented in talks, ran workshops, and organized mixers for the conference attendees. The focus of the event was on social media - how to use platforms such as Twitter as marketing and information gathering tools.

We ended up with around 500 vertices in the graph from the conference by our stopping point. Crawling from November 28th to December 5th we picked up on a substantial amount of chatter regarding the conference the day before it took place and quite a bit on the tail end.

5.1.1 Exploratory Analysis

Being the start of a much larger project we aim to understand how these networks evolve over time. Many of our analyses turn to time series of metrics we might care about in the static case. For this particular work we focus only on these, leaving many interesting dynamic analyses for later study.

5.1.2 Vertex Growth Over Time



We see that many of the vertex additions to the network occur the evening before the conference even begins. This makes some sense; perusing the stream of tweets related to the conference, participants are tweeting about how they are going to the conference the next day, and so want to have their say on the public stream as early as possible.

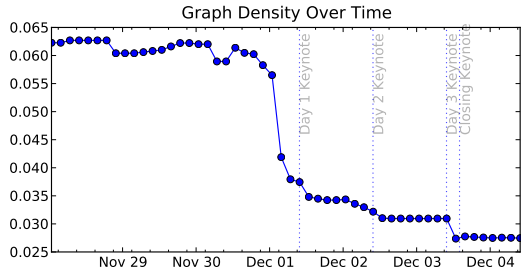
Based on firsthand experience we would expect most vertices would join the network during larger, plenary-type talks such as keynotes. This is indeed what we see here. Node growth also closely mirrors node growth.

We omit discussion of edge growth because it essentially mirrors the exact same trend as vertex growth.

[5]`http://www.online-information.co.uk/online09/ims.html`

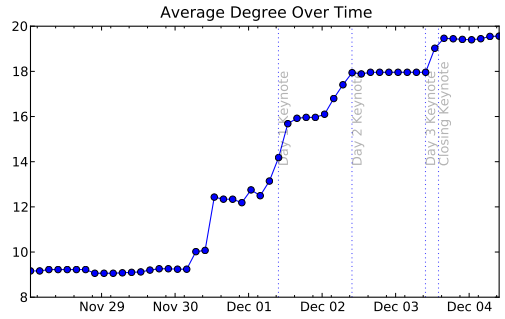
5.1.3 Graph Density Over Time

We notice that the graph density over the length of the conference actually falls. We can interpret it thus: more vertices enter the system than edges get created.

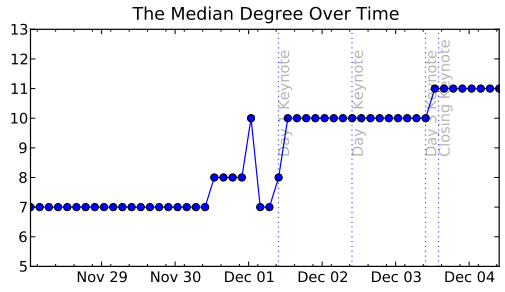


5.1.4 Average Degree Over Time

The average degree over time of the network rises fairly considerably over the course of the conference, from around 14 at the outset of the first keynote to nearly 20 by the last. This would suggest that the conference network becomes more dense over time.



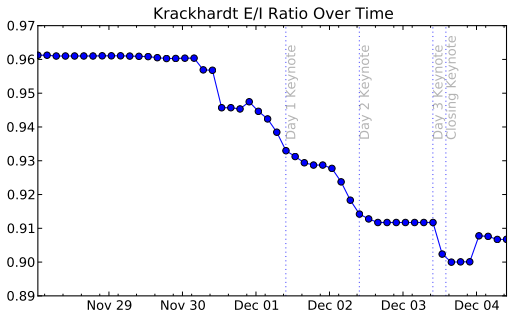
Some pause is in order. Just as likely is the inclusion of highly connected vertices into the network, or a particular vertex forming many more edges than others over the course of the conference. Speakers, for instance, tend to get many new Twitter followers during and right after they give a presentation. Such effects might mask the actual change in the degree distribution. Hence we also include the median degree over time, shown here.



5.1.5 Connectedness to Outside World Over Time

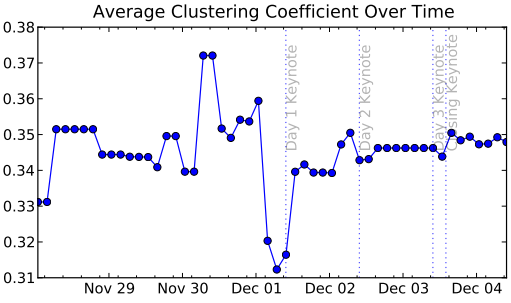
Next we examine Krackhardt's E/I ratio over the course of the conference. Remember that the E/I ratio gives the difference between the percentage of edges that point outward and those pointing inward. The main trend was predictable: The E/I ratio falling by .05 demonstrates that this network forms some sense of a community within this network.

Unfortunately, we have not found a baseline to which we can compare. Should we expect the value to go down more quickly as new nodes are added? Without further work all we can discern from this plot is that the ratio falls, and not if this is more or less than we should expect. Achieving a larger sample of conferences will help us establish a baseline.



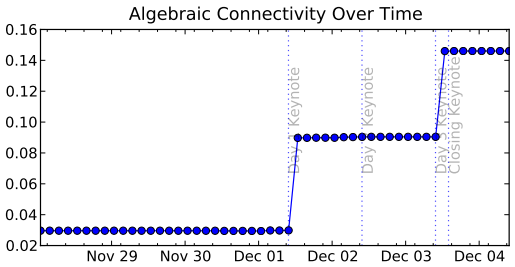
5.1.6 Average Clustering Coefficient Over Time

Here we discuss the average clustering coefficient. Remember that the average clustering coefficient measures the amount of "cliquishness" among the vertices. Aside from some unstable behavior the day before the conference, the average clustering coefficient over the course of the conference remains relatively stable. This suggests that the network does not become any more clique-prone, which we may also infer from the discussion of algebraic connectivity and clustering below.



5.1.7 Algebraic Connectivity Over Time

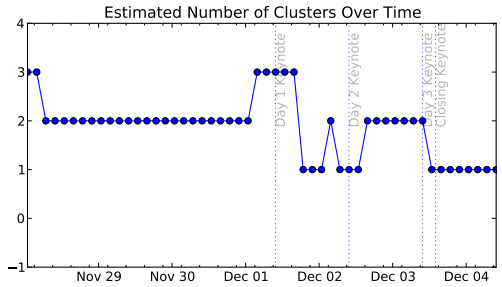
The algebraic connectivity, discussed in section 4.6.1, is the second eigenvalue of the Laplacian graph. It gives us an additional sense of how connected the graph is over time. Striking in this plot is that there is a step-effect that occurs on the first and third keynotes. Examining the values before these massive jumps show that the algebraic connectivity does indeed rise, but by several orders of magnitude smaller than the big steps. The accuracy of this particular metric in our implementation is dubious; given the difficult nature of estimating the algebraic connectivity of a network that contains many isolated vertices and components the metric might not be meaningful.

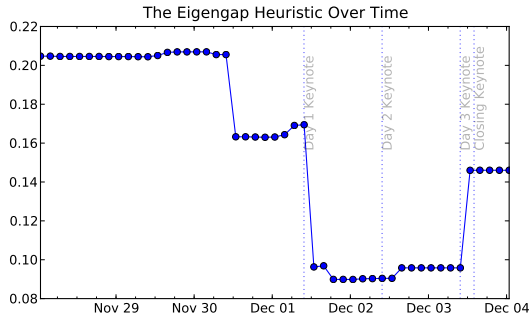


5.1.8 *Possible Clusters Over Time and the Eigengap Heuristic*

Here we use von Luxburg's eigengap heuristic [11], along with our decision rule outlined in section 4.6.2 to estimate the total number of possible clusters. We plot both the estimated number of clusters and the eigengap heuristic to give a sense of how the two are related.

It is worth noting that the eigengap heuristic goes down considerably over the course of the conference, and the timing of the changes, as with the algebraic connectivity, coincide with keynote addresses. As with the E/I ratio, it is difficult to discern how connected the network actually becomes without adequate baselines.





6. CONCLUSIONS, RESERVATIONS, AND EXTENSIONS

With the explosion of public APIs researchers can analyze in real-time network growth and interactions. We presented a new software toolkit that dynamically pulls such networks off of Twitter and provides simple, rich classes for dealing with these dynamic networks. From a data-collection perspective, Social Aviarygives researchers the tools to pull new data from events as they happen. From a library perspective, Social Aviaryalso gives developers tools in the familiar NetworkX paradigm to test their own models. Finally, developers and researchers also get a set of rich exploratory data analysis tools to accompany the others.

As a first step in a much larger project, we analyze one conference. The sheer number of questions to answer with a data set of interactions, along with the timing of vertex and edge creation, is dizzying. Yet we understand that we need both real baselines and more conferences with which to compare, so we can gain some intuition about what we ought to expect from these sorts of dynamics.

Regarding baselines for exploratory data analysis, we are currently researching three. The first is obvious: as we collect more conferences we will observe common trends, and can begin to compare trends to each other. Second, we hope to establish a number of theoretical bounds to some of the metrics of interest. What precisely is appropriate from a theoretical perspective is not obvious, given that the conference represents an oddity in normal social network theory. Third, we hope to simulate based off of some existing models, once we understand how to extend them to this type of dynamic.

7. REFERENCES

[1] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98), 2009.

[2] Santo Fortunato. Community detection in graphs. *ArXiV*, 2009.

[3] A. Jamakovic and S. Uhlig. On the relationship between the algebraic connectivity and graph's robustness to node and link failure. *Working Paper*, 2003.

[4] Akshay Java, Xiaodan Song Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, 2007.

[5] Gueorgi Kossinets and Duncan Watts. Empirical analysis of an evolving social network. *Science*, 311, January 2006.

[6] David Krackhardt and Robert Stern. Informal networks and organizational crises: An experimental simulation. *Social Psychology Quarterly*, 1988.

[7] Jure Leskovec, Jon Kleinberg, and Christos Foloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1, March 2007.

[8] Jure Leskovec, Kevin Lang, Anirban Dasgupta, and Michael Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *arXiv*, 2008.

[9] Bojan Mohar. The laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 1991.

[10] Hamilton Ulmer and Ryan Noon. Tiresias and social viary: A comprehensive framework for describing, analyzing, and visualizing connectivity data over time.

[11] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[12] Duncan Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 1998.