THE EXPRESSION OF HUMAN BEHAVIOR IN ONLINE NETWORKS

Jacob Ratkiewicz

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Department of Computer Science
Indiana University
May 2011

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

_____

Filippo Menczer, Ph.D.

_____

Alessandro Flammini, Ph.D.

_____

Steven Myers, Ph.D.

_____

Alessandro Vespignani, Ph.D.

May 2011

ACKNOWLEDGEMENTS

This work was made possible by encouragement and support from many people.

My endlessly patient advisor, Filippo Menczer, has spent countless hours answering my questions, giving me direction, and reading my manuscripts. Without his help, support, and belief in me, I could never have even begun to write this document. I am also grateful to the other members of my committee — Alessandro Flammini, Steven Myers, and Alessandro Vespignani. Thank you all for sticking with me through the twists and turns my graduate career has taken, always with a gentle push in the right direction when I needed it.

Thanks are also due to all of my collaborators — Filippo Menczer, Alessandro Flammini, Michael Conover, Bruno Gonçalves, Alessandro Vespignani, Santo Fortunato, Markus Jakobsson, Snehal Patil, and Matthew Francisco. I am very grateful for the insights, patience, and dedication they have shared with me over the years.

The Networks and Agents Network (NaN) group has also been a valuable resource for development of new ideas, and a friendly place to practice presentations. I feel fortunate to count its members as my friends as well as my colleagues.

Rob Henderson, Bruce Shei, and the rest of the systems support team in the Computer Science department have been tremendously helpful.

Ciro Cattuto, Vittorio Loreto, and Massimo Marchiori were the sources of much valuable assistance in designing the research performed in Chapter 6.

I am also grateful to Mark Meiss for the Indiana University traffic data, to Ricardo Baeza-Yates, who facilitated access to the TODOCL dataset, and to Virgil Griffith, who helped facilitate an (ultimately unsuccessful) attempt to access data from the Internet Archive.

Thanks for generous support goes to the ISI Foundation in Turin, Italy. Support for portions of this work also came from NSF Grant No. IIS-051365 and NSF Grant No. IIS-0811994.

Thanks to my undergraduate advisor, James Wolfer, for getting me interested in research.

My friends Jeremy Engle, Ben Markines, Mira Stoilova, and Mike Conover were the source of much encouragement and support during my graduate school career, as was my family.

Thanks to everyone who helped me relax during the process of writing this dissertation with a quick game of foosball, especially Nicola 'The Eagle' Perra.

Finally, I am grateful to Laura Brunetti for all of her patience, encouragement, and understanding.

Jacob Ratkiewicz

THE EXPRESSION OF HUMAN BEHAVIOR IN ONLINE NETWORKS

The wide adoption of Web 2.0, in which users can interact with Web sites to generate new content, has a serendipitous side effect. All of this user-generated data provides researchers with a unique lens on the behavior of the users who created it. While instrumenting millions of users with a device that records everything they read in real life would be impossible, we can easily record the articles they read on Wikipedia. Similarly, we can use Twitter data to map the interactions between tens of thousands of people, as well as studying the topics they discuss.

I outline several studies taking advantage of this trove of behavioral data. Initially focusing on Wikipedia, I examine the patterns in the paths that users take when navigating from article to article, and contrast these with similar data for several other large Internet destinations. I then develop an understanding of bursty popularity dynamics, discovering that bursts in the attention to a page have dynamics similar to that observed in natural phenomena, like earthquakes and avalanches; I also present a simple model able to capture these dynamics. Next I switch gears — away from looking at users as they travel between topics, and towards looking at how *topics* (memes) travel between *users*, and how users interact with each other. I frame this research in the context of political discussion on Twitter. I first perform a general overview of the space of this discussion, examining how users connect with each other. I conclude with a case study, the Web site `truthy.indiana.edu`, which focuses on the case of the deceptive dissemination of ideas, or so-called *astroturf*.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Overview and Themes

In the early days of the Web, content creation was mainly the priviledge of a small fraction of Web users. These were the privileged few who were skilled in HTML and had access to Web hosting space, both a necessity for creating and sharing information on the Web. As the Web matured, the nature of content creation underwent a process of democratization — from early Geocities[1] free web hosting to today's social Web media such as Wikipedia[2] and Flickr.[3] These technologies reduced the barriers to creating Web content by obviating much of the need for technical skill or expensive Web hosting.

Today, the average Web user may not even be aware of this shift. We create content in the modern Web almost without thinking about it — rating movies on Netflix, editing a Wikipedia page, or posting a product review on Amazon. The rise of these large online social systems is a windfall for researchers in (at least) two major ways. The first is the fact that these systems by their very nature aggregate data of a similar type and under a consistent data model. Before Flickr, for example, a researcher interested in studying how people share images online might have to do a

[1] `geocities.yahoo.com`

[2] `www.wikipedia.com`

[3] `www.flickr.com`

large-scale crawl of the Web at large, and deal with images in a large variety of formats and with inconsistent, or non-existent, metadata. The second major benefit is that users create two kinds of information when interacting with these online social systems. The first is the data they actually use the system in order to create — pictures and annotations on Flickr, or articles on Wikipedia. The second kind of information is data about *how they use* the systems in question when creating and consuming the first kind of information.

While tracking the interactions of humans in the real world is not practical on a large scale, when those interactions are reflected in a large social system on the Web they may be captured and analyzed. It would be impossible to instrument millions of people and record the title of every book they read, but we can track the reading habits of Wikipedia users. We cannot record people's conversations, but we can track what they choose to share on Twitter, and with whom they choose to share it. The study of these rich Web data sets is really a proxy to studying the behavior of the real people whose actions create and shape them. How, then, do people navigate these information networks? What influences a user as she chooses the next page she will visit, the next hyperlink she will click? We know that linked pages are more likely to be similar to each other, both in their content and in their neighborhoods, than pages chosen at random [**Men04**]. How does this affect browsing behavior?

An important area of research in network science is that of creating models that can capture the evolution of networks. Many such models have been proposed for information networks such as the Web, each able to reproduce some facets of its structure. The recent availability of large-scale longitudinal datasets containing the growth of real-world information networks gives us the opportunity to actually validate these models — not just on how well they reproduce a system's final state, but also on how well they capture the changes the system undergoes on the way. In other words, we can ask how well present models capture the *dynamics* of the growth of information systems, in contrast with their end state.

Models for network growth can also be thought of as models for the popularity of a set of networked items, as the in-degree, or number of edges pointing to an item, is often considered a measure of popularity. Another way to view the popularity of Web pages or Wikipedia articles is as the accumulated attention of all the people who have navigated to them. Does this attention behave

in a smooth way? Are there universal regularities? Are changes driven mainly by endogenous events, or externally?

Just as Wikipedia presents a network of *ideas* along with people travel, social blogging sites such as Twitter present a network of *people* who communicate *ideas* to each other. This allows us to study the popularity of an idea, or *meme* more richly — whereas in Wikipedia we might only know that a page was visited some number of times, on Twitter we can tell exactly *who* has been exposed to a particular meme, from whom they first learned about it and when, as well as to whom they communicated it. What can we tell from this about the way that ideas are spread? Certainly some users are more amenable to spreading some ideas (perhaps those they agree with) than others. Can we use these transmission patterns to determine users' interests, and find communities? Finally, can we identify attempts to maliciously insert memes and make them appear to represent widely held opinions?

1.2. Outline

I begin by providing some background information in complex networks analysis, document modeling, and a little machine learning, all in Chapter 2. I then overview various works related to portions of mine in Chapter 3. Chapter 4 then describes the data that I use in the rest of the paper. Each of these data sets captures a different facet of human online behavior.

Having laid the foundation for the work to follow, I begin with some preliminary experiments in Chapter 5. These experiments focus on Wikipedia, and were motivated by a desire to understand the basics of how users navigate between topics of Wikipedia. More than just understanding Wikipedia itself, the results here generalize to some extent in understanding how a user's attention moves between topics of interest. As a step towards examining this generalization, I also present results of some comparisons with other large networks.

Chapter 6 presents some experiments focusing on the nature of the change over time of popularity — the aggregate measure of users' attention. In particular, I show that this change is sometimes dramatic — and that these dramatic changes share some characteristics with natural phenomena, like earthquakes and avalanches. The experiments here are performed on large-scale longitudinal data from Wikipedia, and the web space of a country. I evaluate the performance of a standard model [**BA99**] in capturing these dynamics, and find that it does not perform well. I

conclude by discussing a new model [**RFF$^+$10**] which is able to reproduce the several key features of the network, including the sudden shifts in attention present.

In Chapter 7, I change focus from users' impact on the topics they view, to the impact of the ideas themselves on the users that consume them, and how these ideas travel between users. This exploration is structured as a number of experiments performed in the context of political discourse on Twitter. A central finding here is that the connections between users tell us a lot about the user's political alignment — users are likely to form certain kinds of connections only to other users with whom they agree.

Finally, in Chapter 8, I present a case study tying together many of the themes addressed here. This is in the form of a software system and associated Web site, collectively called 'Truthy,' for detecting a certain kind of abuse on Twitter: namely, the spreading of *astroturf*. Distinct from spam, astroturf is a scheme to create a false sense of community consensus about a topic; it is so named to contrast it from a true 'grassroots' effort. The system performs detection and tracking of tweets from a stream of raw data from Twitter, identifying first topics about American politics, then topics of general interest in this space, then finally topics which may represent astroturfing attempts. I find that a simple off-the-shelf classifier is able to achieve very high accuracy in identifying these astroturf memes (which I call 'truthy' memes), using features from the diffusion network of these memes.

I conclude in Chapter 9, summarizing the major results and outlining some directions for future work.

CHAPTER 2

BACKGROUND

Here I provide a summary of the theoretical techniques that I use throughout the rest of this work. Techniques are presented here if they are used in my work in a more or less 'off the shelf' way. Work upon which I build or improve, and work similar in focus to mine, is discussed in the chapter on related work (Chapter 3).

2.1. Graph theory

2.1.1. Undirected graphs

Much (if not all) of the results described here rely on *graph theory*. Simply put, a graph is described by a set V of *vertices* (also called *nodes*), together with a relation $E \subseteq V \times V$ which relates these vertices; E is often called the *edge set* or the *edge relation*. When this edge relation is symmetric, so that

$$\forall v_1, v_2 \in V \left((v_1, v_2) \in E \iff (v_2, v_1) \in E \right), \tag{1}$$

we refer to the graph as being *undirected.*

Examples of objects which can be modeled as undirected graphs include the following:

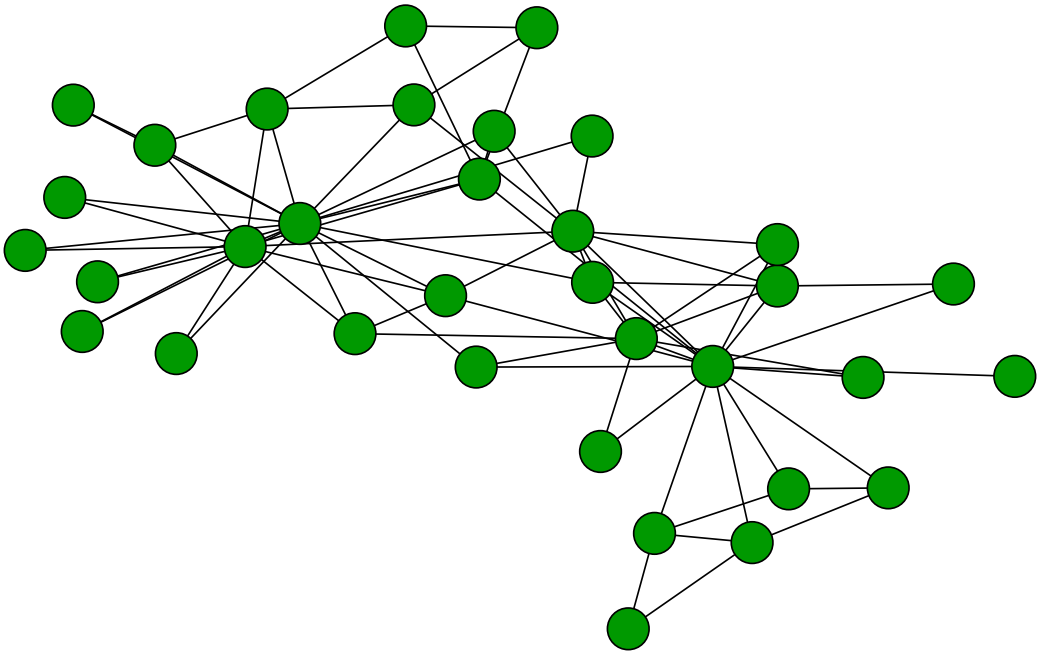


FIGURE 2.1. Network of friendships between members of a U.S. university karate club [**Zac77**]. This network is often referred to as the *Zachary karate club network*.

The U.S. highway network: Cities can be thought of as vertices, with highways the edges that connect them. Note that since there is no such thing as a highway only passable in one direction, it is fair to think of this network as undirected.

The interactions between proteins in a cell: The proteins themselves are the vertices, with two proteins being connected if they interact with each other to perform some biological function.

Social interactions between people: If we make the comforting assumption that person A calling B a friend implies that B feels similarly about A, we can model friendships between people as an undirected graph. Such a graph, in which the nodes are people and the edges reflect some relationship between the linked people, is often called a *social network.* The edge relation may be variously thought to model real-life friendship, romantic involvement, physical proximity at a certain distance threshold, or any number of other relationships.

Shown in Figure 2.1 is an example of a social network. This graph represents the friendships between members of a U.S. university karate club [**Zac77**]. Here, of course, vertices are people, and two people are linked if they reported to the researcher that they were friends.

2.1.2. Directed graphs

When the edge relation E of a graph (V, E) is not constrained to be symmetric, the graph is called a *directed graph* or *digraph*. Note that any undirected graph can be thought of as a directed graph (in which the edge relation just happens to be symmetric). The following are some simple examples of directed graphs:

City intersections: City intersections may be modeled as vertices, with streets connecting them. The existence of one-way streets in real life means that this graph must be a directed one.

A finite-state automaton: This theoretical computer science concept is really a graph, in which the states of the machine are vertices and state transitions are encoded by directed edges between states.

A network of Web pages: In this abstraction, Web pages are vertices and two pages A and B are connected by a directed edge (A, B) if page A contains a hyperlink to B.

Social networks, too, can be thought of as directed, for social relationships that are not symmetric. For instance, the organization chart of a company can be thought of a directed social network — people are vertices, and the edge (A, B) is present just when B is A's supervisor.

2.1.3. Weighted graphs

Suppose that we wish to refine the U.S. highway graph given as an example in 2.1.1, by capturing the fact that some cities are closer together than others are. To do this, we might attach to each edge a number, encoding the *cost* attendant to following that edge; this value may also be thought of as a *distance.* In the case of a directed graph, we can imagine that the distance from A to B might be different than the distance in the opposite direction (maybe one way is uphill!) A graph to which real values have been attached to the edges (by some function $W : E \to \mathbb{R}$) is referred to as a *weighted graph*. We can imagine instances of both *directed* and *undirected* weighted graphs.

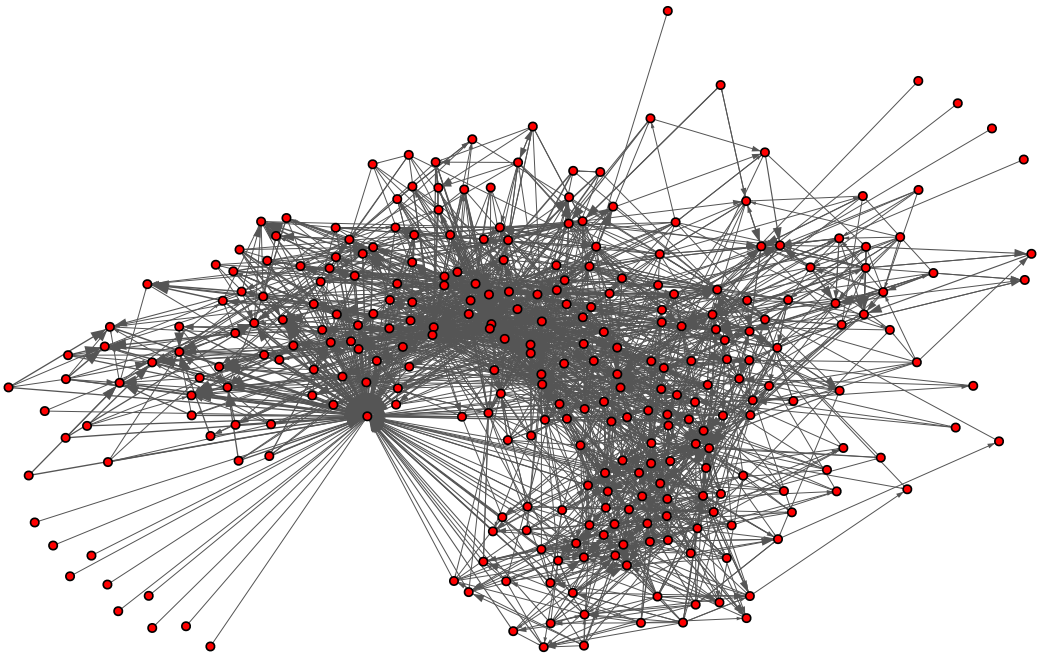


FIGURE 2.2. Network of neural connections in *C. elegans*, a type of nematode often used as a model organism [**WS98**].

In some contexts, the edge weight is also thought of as a *similarity* between the two linked items, or the strength of a connection. Figure 2.2 shows a network of the connections between neurons in *C. elegans*, a type of nematode often used as a model organism. The network is directed and weighted, with edge directions represented by arrows, and edge weights by the thicknesses of the edges. However, the size of the network makes these hard to discern. I will go on to introduce some analytical techniques which make it easier to understand networks which are too large to visualize completely.

The following are some other examples of weighted graphs (besides the highway example given earlier):

Distance between cities: In general, systems where there is a concept of distance lend themselves well to modeling by weighted graphs. One can imagine a network where the nodes are U.S. cities, and all pairs of cities are connected by an edge weighted by the straight-line distance between them. Such a network would be a weighted, undirected network — undirected because if city B is 50 miles from A, as the crow flies, the same is true in the

other direction. (Since all pairs of cities are connected, this would also be an example of a *complete* graph.)

A network of Web pages — with traffic: Consider again a network where the nodes are pages, and two pages are connected by a directed edge if one contains a hyperlink to another. This network, given earlier as an example of a directed network, can have weights affixed to its edges as well. I will later explore some questions related to networks of this type, where the edge weights are derived from the number of users who have followed a particular link.

As may be apparent from the visualization of the *C. elegans* neural network in Figure 2.2, many real-world networks are large enough that simply visualizing them in their entirety is not very useful. Fortunately, there exist a number of analytical techniques for quantifying various facets of a network's structure. I will discuss a few of those in the following sections.

2.1.4. Degree and strength

One of the measures that may be considered in analyzing a network is the *degree* of its nodes. The degree of a node in an undirected network is simply the number of edges that connect to it, divided by two (to account for the fact that an undirected edge is represented by two directed edges). Thus, a network with a high average degree is one in which the nodes are densely connected. In a directed network, we can count separately the number of edges going into a node and coming out of it; we refer to these counts as the *in-* and *out-degree* of the node, respectively. For large networks, these values are often aggregated in histograms as the *degree distribution* of the network in question. The symbol k is often used to represent the degree of a node, with k_{in} and k_{out} representing its directed in and out-degree, respectively. Along with looking at the degree of a particular node, we can also consider its *strength*, which is the sum of the weights of the edges adjacent to it. Just as with degree, we can consider the *strength distribution* for the nodes in a network. The symbol s often refers to the strength of a node, with s_{in} and s_{out} being used for in- and out-strength, respectively, for a directed network. Figure 2.3 shows histograms of the distributions of degree and strength for the *C. elegans* neural network shown in Figure 2.2. Note that these distributions are heavily skewed and have very long tails, especially in the in-degree and in-strength cases. I will later explore some plotting techniques to produce more useful histograms for such heavy-tailed distributions.

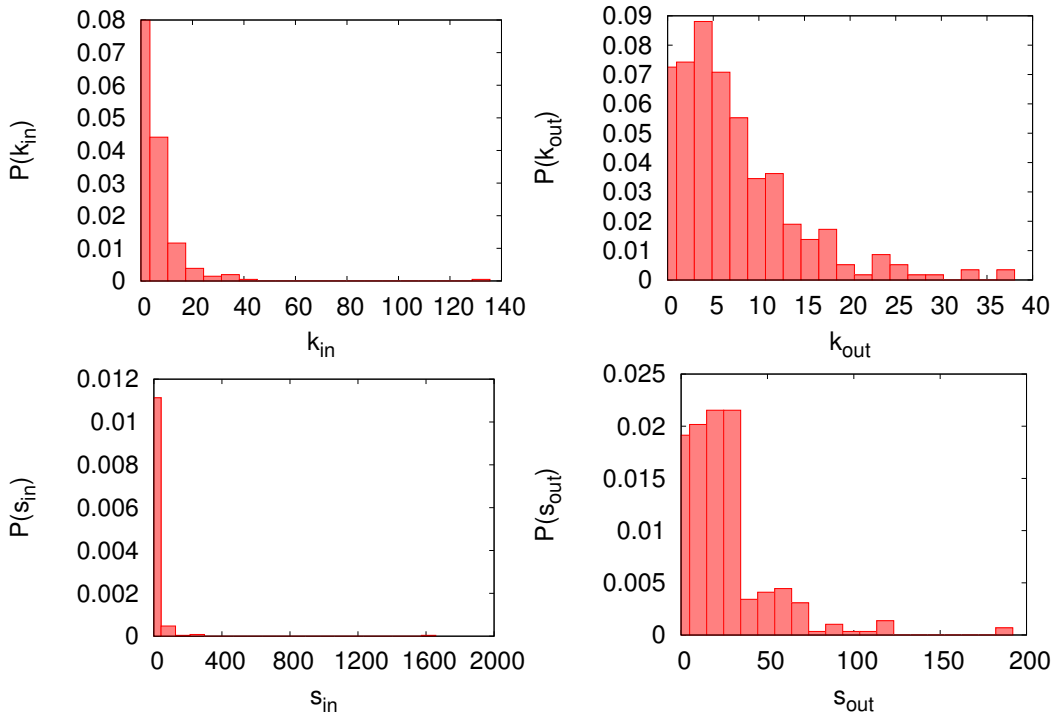


FIGURE 2.3. Distributions of in degree (top left), out degree (top right), in strength (lower left), and out strength (lower right) for the *C. elegans* neural network.

2.2. Graph clustering

An important problem in studying networks is the division of their nodes into related groups, or *clusters*. While algorithms exist that produce overlapping clusters, I focus here on methods that result in a partition of the graph. This division often has the goal of placing nodes together that are similar in some way, often in terms of their connections to other nodes. Here I first describe a quantitative tool, *graph modularity*, which helps us recognize a good clustering when we see it; I then describe several methods for developing these partitions, as well as some of their advantages and drawbacks.

2.2.1. Evaluating clusters

A partition of a given graph has an associated metric called *modularity* [**New06b**]. Modularity, often denoted Q or q, is defined for graphs with positive integral edge weights as follows. For an undirected network with n nodes connected by m edges, define the *adjacency matrix* $\mathbf{A}$ such that $\mathbf{A}_{ij}$

is the weight of the edge connecting the vertices i and j, or 0 if no such edge exists. Suppose the nodes are partitioned among some arbitrarily-numbered clusters, and let the *cluster assignment* c_i be equal to the index of the cluster into which vertex i is placed. Recall that k_i denotes the degree of the vertex i, and let m be the number of edges in the entire network. Then

$$Q = \frac{1}{2m} \sum_i \sum_j \left[\mathbf{A}_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{2}$$

where

$$\delta(c_i, c_j) = \begin{cases} 1 & \text{if } c_i = c_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

is the Kronecker delta. The intuition behind this measure is to give a higher score to clusterings which accomplish a higher degree of intra-cluster links than would be expected in a completely random network with the same degree sequence. Though modularity is often used and works well in practice, one of its drawbacks is that it depends on the size of the network; thus, it's not fair to compare the modularities of networks of different sizes. (In Chapter 7 I introduce a technique to make this comparison.)

2.2.2. Clustering algorithms

A plethora of algorithms exist for clustering nodes in a graph, each with its own set of advantages and disadvantages [**Sch07**]. Here, I explore the two algorithms that I use later, namely Newman's leading eigenvector method [**New06a**] and the label propagation method of Raghavan *et al.* [**RAK07**]. Note that in using each of these algorithms, I supply the number of desired clusters as an input parameter.

2.2.2.1. Leading eigenvector. Given that modularity is a goodness measure for clustering assignments, it is perhaps intuitive to design a clustering method that optimizes it directly. This is impossible to directly do efficiently in the worse case, however., The leading eigenvector method, then, is an example of an algorithm that attempts to efficiently *approximate* a maximization of modularity. In its essence, it recursively determines splits of a network (or subset of a network) into two clusters, in such a way as tends to maximize the modularity of the network under the chosen split. This is done by building a *modularity matrix* which represents the potential modularity of the network under various splits. The eigenvector corresponding to the most positive eigenvalue of this matrix

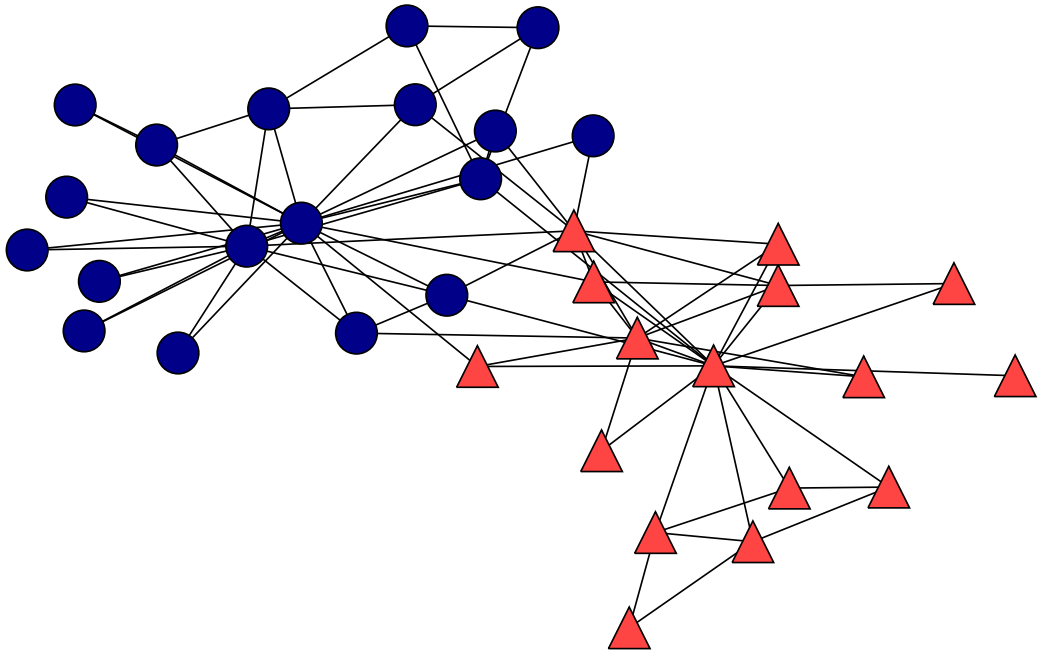


FIGURE 2.4. The Zachary karate club network, partitioned into two clusters according to Newman's leading eigenvector method. The colors and shapes of the nodes reflect the assigned clusters. The modularity of this clustering is 0.371.

can then be used to assign nodes in one of two clusters depending on the signs of its elements; further recursive splits can be performed by iterating this method (with some refinements). Figure 2.4 shows the Zachary karate network, clustered by this method; the color of the nodes reflects the cluster to which they are assigned.

This method has the advantage that it is relatively easy to compute, and fast; properties of the modularity matrix make it possible to compute multiplications in time $O(n + m)$ for a network with n nodes and m edges. However, finding eigenvectors in a matrix remains slow, requiring time $O(n^2)$ in the best case, where only a few eigenvectors are desired (as for a fixed number of splits), and time $O(n^3)$ when the algorithm is meant to split the network as much as possible.

2.2.2.2. Label propagation. This method does not attempt to optimize modularity directly; it eschews global metrics of cluster goodness like modularity in favor of local optimizations. Intuitively, this algorithm works by initially assigning nodes to clusters arbitrarily, then iteratively assigning each node to the same cluster as the majority of its neighbors, until convergence is reached. New

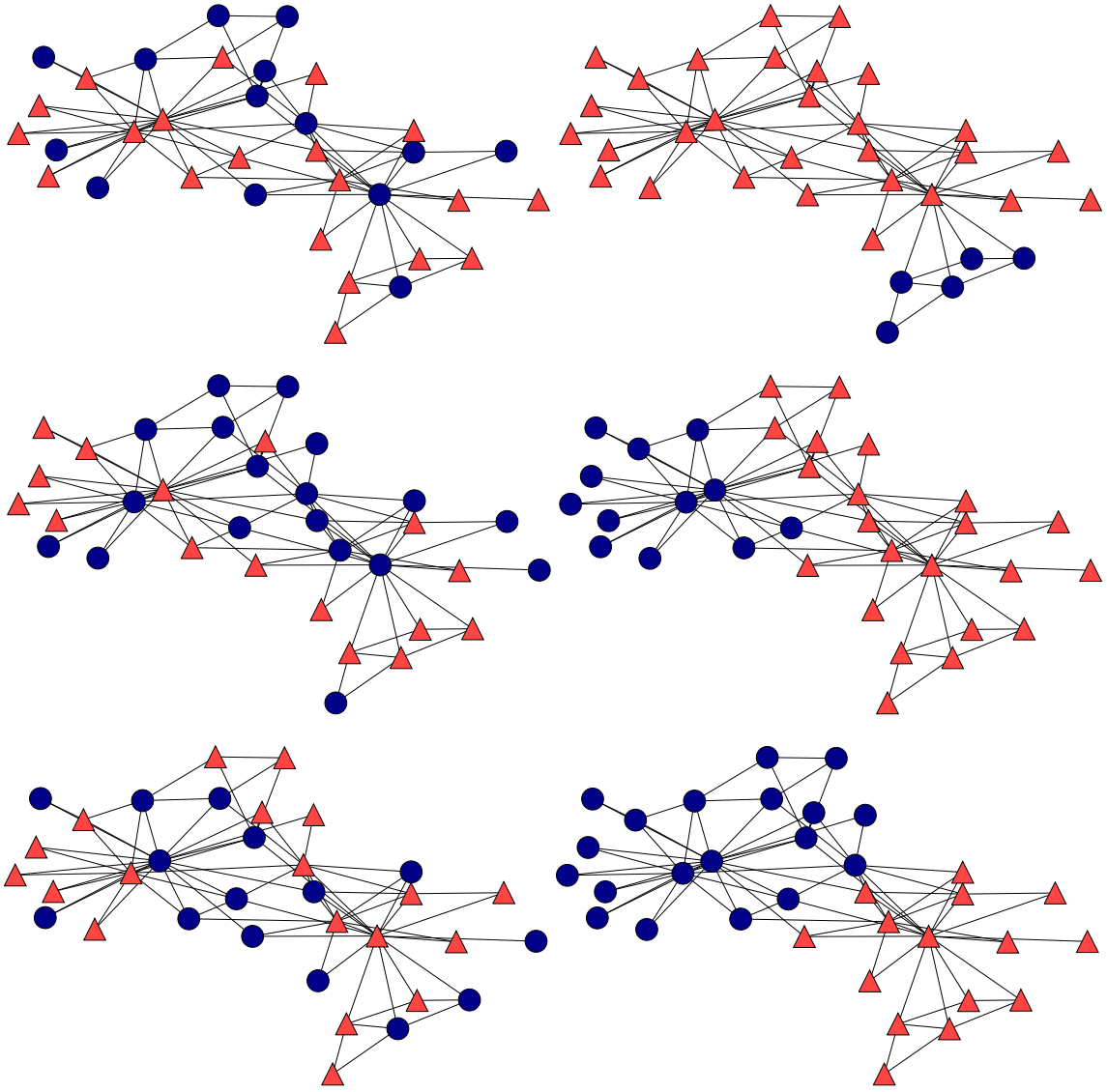


FIGURE 2.5. Three initial random cluster assignments of the nodes in the Zachary karate network (left), and the node assignments after convergence of label propagation clustering given those assignments (right). Widely varying results are possible, including very poor results for unlucky starting conditions.

clusters for nodes do not take effect until the next iteration of the algorithm, so there is no restriction on the order in which they are computed. In the case of ties, where a node does not have a majority of neighbors in any one cluster, the new label for the node is chosen randomly among the tied clusters. It is from this mechanism of propagating labels to nodes from the majority of their neighbors that the algorithm gets its name. This method has the advantage of being very fast, running in time $O(m)$ for each iteration. The authors observe that five iterations is usually enough to cause the algorithm to be very near convergence in some real-world examples; however, they do not prove convergence results in general.

One weakness of this method is derived precisely from the same source as its main strength — since it does not use global measures of goodness, it is prone to getting stuck in local maxima. This weakness is made manifest in the sensitivity of the algorithm to the initial arbitrary assignment of vertex labels. A bad or unlucky initial assignment can cause problems in two ways: it can doom the algorithm to inevitably become stuck in a local maxima, or it can be such that two runs of the algorithm from the very same starting conditions result in wildly different cluster assignments (due to the randomness involved in breaking ties). Figure 2.5 shows the Zachary karate network again, with three initial random cluster assignments on the nodes as well as the cluster assignments that result from running the label propagation algorithm given those initial assignments. Note that a high degree of variation is possible, including some very poor results. I revisit this problem, and suggest a fix that involves combining these two clustering methods, in Chapter 7.

2.2.3. Evaluating the similarity between two cluster assignments

The issue of stability touched on in the previous section raises an important question: given two cluster assignments for the same graph, how can we quantify the degree to which they agree? In answering this question, we must first define 'agreement.' For a graph G with n nodes, we say that two different cluster assignments C and C' 'agree' for the nodes a and b if they both either put a and b in the same cluster, or put a and b in different clusters. This notion of 'agreement' is expanded into a similarity measure known as the *Rand index* [**Ran71**], which can be summarized as follows. Arbitrarily number the clusters of C as $\chi_1 ... \chi_I$, and likewise number the clusters of C' as $\chi'_1 ... \chi'_J$. Define then the *contingency matrix* $\mathbf{N}$, where the i, j entry of $\mathbf{N}$ is the number of nodes of G simultaneously in χ_i and χ'_j. Further define the row sum $a_i = \sum_j \mathbf{N}_{i,j}$, and similarly the column

sum $b_j = \sum_i \mathbf{N}_{i,j}$. The Rand index may then be expressed as

$$\mathrm{R}(C, C') = \frac{1}{\binom{n}{2}} \cdot \left[\binom{n}{2} - \left[\frac{1}{2} \left(\sum_i a_i^2 + \sum_j b_j^2 \right) - \sum_i \sum_j \mathbf{N}_{i,j}^2 \right] \right] \tag{4}$$

This measure is in the range $[0, 1]$. It is 0 for two cluster assignments that disagree over the placement of every node (e.g. an assignment that places all nodes in the same cluster, and another that places each node in its own cluster). It is 1 for two cluster assignments that are in perfect agreement. The paper cited above provides a nice example making more clear the intuition behind this measure. Of course, two uniformly random cluster assignments will be very unlikely to have a Rand index of either zero or one; further, the value of the Rand index for two random cluster assignments depends on the size of the cluster assignments, and the size of the graph. Since it is useful to have an index which is zero when two clusterings agree with each other at chance levels rather than not at all, another often-used measure is the Adjusted Rand Index (ARI) [**HA85**], which has this property. Given the definition of the contingency matrix and its row and column sums given above, the adjusted Rand index may be expressed as

$$\mathrm{ARI}(C, C') = \frac{\sum_{ij} \binom{\mathbf{N}_{i,j}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}. \tag{5}$$

This measure takes on values in the range $[-1, 1]$. 1 indicates perfect agreement with -1 indicating perfect disagreement; values near 0 indicate agreement at chance levels.

2.3. Graph growth models and random graphs

This section provides a basic overview of the concept of a network growth model and random graphs in general, with some simple examples. See 3.2 for a description of some specific growth models in more detail.

It is often useful to produce a random graph, for instance for comparison with a real-world graph to see if some observed properties in the real-world graph differ significantly from what one would expect by chance. Perhaps the simplest possible random graph is the Erdős-Rényi (ER) random graph [**ER60**]. Such a graph is constructed based on two parameters — some number of nodes n, and a *link probability* p, being the probability that any two nodes are connected. Alternately, an absolute number of links m can be specified in place of p, in which case m edges are sampled at random from the set of all possible edges. Note that the probability of connecting two nodes, in either

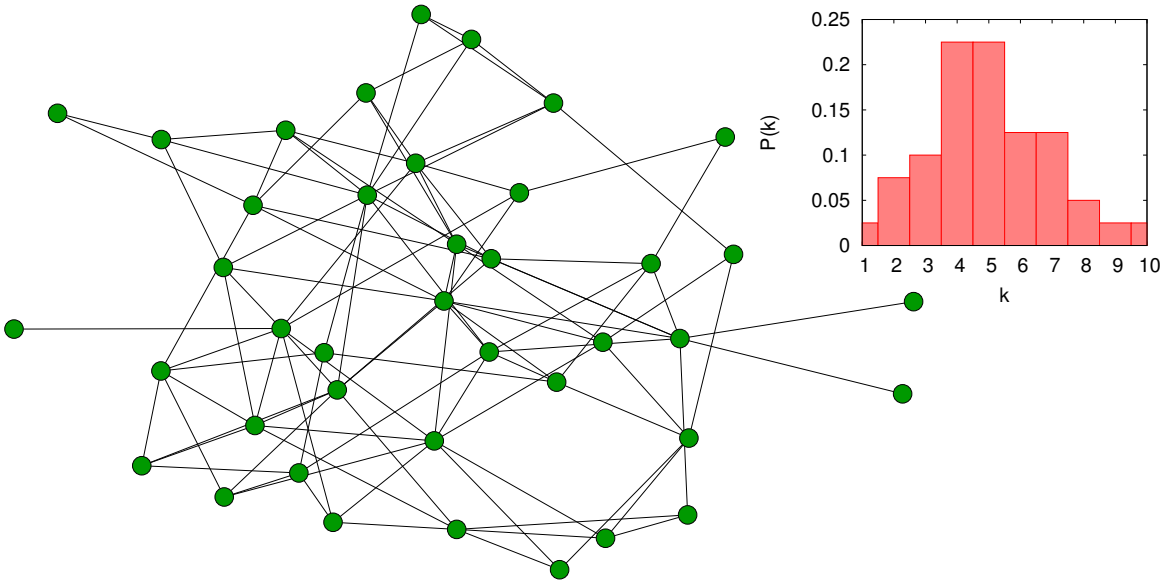


FIGURE 2.6. An Erdős-Rényi random graph, with number of nodes $n = 40$ and number of edges $m = 100$. The inset contains the degree distribution of the network; note that it is peaked, rather than skewed.

case, is independent of any other connections present in the graph. Random graphs constructed by this method are not likely to exhibit many of the properties present in real-world networks, such as a broad distribution of degree; while in real-world networks there are likely to be nodes with a large number of connections, all nodes in ER random graphs have about the same number of connections (namely $n \cdot p$). Figure 2.6 shows an Erdős-Rényi random graph with $n = 40$ and $m = 100$, with the degree distribution of the graph in the inset. Note that the degree distribution is peaked, rather than the skewed distribution we will observe in many real-world graphs.

The Erdős-Rényi random graph model is not an instance of a growth model, as there is no allowance for the iterative addition of nodes. Such models are often used to model the evolution over time of real-world networks. Such models generally start with some initial state of a network, then iteratively add nodes and connect these nodes to existing nodes in the network based on properties of those nodes. Certain choices of these properties can yield graphs with similar properties to those found in real-world graphs. I describe a number of these growth models in 3.2. Part of Chapter 6 describes a novel growth model designed to capture the dynamics of bursts of online attention.

2.4. Document modeling

A number of problems in information retrieval are hinged on a good definition of the documents involved. The concept of a 'document' is sufficiently broad to encompass many objects which can be though of as sets of words. For example, the following can all be thought of as documents:

- A Wikipedia article
- A Web page
- The set of tweets posted by a particular Twitter user
- The set of category tags associated with a Wikipedia page
- The set of emails sent by a particular user.

In all of these contexts, we often refer to the set of all documents as the *document corpus* or simply the *corpus.*

2.4.1. The vector space model

A useful model for documents, which allows them to be manipulated mathematically, is the *vector space* model. In this model, documents are vectors in multidimensional Euclidian space. The number of dimensions of these vectors is equal to the number of distinct words in the entire corpus; thus, all document vectors have the same dimensionality. The value corresponding to a word w in a document D's vector space representation is determined by w's *term frequency* in D, and sometimes by its *inverse document frequency*, as follows:

2.4.1.1. Term frequency. The term frequency $\mathrm{TF}(D, w)$ of word w in document D is, in the simplest case, the number of times that the word w appears in D. However, it is sometimes desirable to normalize this value by the length of the document D itself, so that $\mathrm{TF}(D, w)$ instead represents the fraction of words in D that are w. Other, more complicated, normalizations are also sometimes used.

2.4.1.2. Inverse document frequency. Words that appear in most of the documents in a corpus (such as, for example, *the*), do little to distinguish one document from another. It is therefore sometimes useful to encode the fact that a word is common, using the *inverse document frequency*, or $\mathrm{IDF}(w)$. This measure does not depend on a particular document; rather, it aims to give low weight

to terms which appear in many of the documents in the corpus, with higher weight to those that appear in only a few documents. One common formulation of the IDF is

$$\text{IDF}(w) = \log\left(\frac{1+|\mathbb{D}|}{|\mathbb{D}_w|}\right), \tag{6}$$

where $\mathbb{D}$ is the set of all documents, and $\mathbb{D}_w$ is the set of documents that contain the word w. Thus, when $\mathbb{D}_w \approx |\mathbb{D}|$, this value approaches 0; when $\mathbb{D}_w$ is small, the value grows.

Combining TF and IDF is often a useful technique for constructing the vector-space representation of a document. A common formulation for this, called TF-IDF, to represent the component of a document D's vector for word w as

$$\text{V}(D, w) = \text{TF}(D, w) \cdot \text{IDF}(w). \tag{7}$$

Thus, the high values of $\text{V}(D, \cdot)$ will be for words w that appear often in D and do not appear in many other documents; thus, those words that are most useful for distinguishing D from other documents. The value of $\text{V}(D, w)$ will be 0 for words w that do not appear in D. We can imagine, then, that the vector-space representation of many real-world documents will be sparse, as many real-world documents do not use a significant fraction of the words in the English language.

I use the term 'document vector' to refer to the vector-space representation of a document, irrespective of how this vector space representation is constructed (via TF-IDF or something else).

2.4.2. Cosine similarity

The vector-space model makes possible a convenient method for computing the similarity of two documents — computing the angle between their vectors. For two document vectors D_1 and D_2, this is given by

$$\sigma(D_1, D_2) = \frac{D_1 \cdot D_2}{||D_1|| \cdot ||D_2||} \tag{8}$$

Being the cosine of an angle, this measure takes on values in the range $[0, 1]$ — 0 when the two documents contain no words in common, and 1 when they are identical. It is widely accepted that when two documents of sufficient length have a high cosine similarity between their TF or TF-IDF vectors, they are likely to be semantically similar as well [**vR79**].

2.5. Evaluation techniques

This section overviews some techniques for evaluating the quality of algorithm results.

2.5.1. Precision and recall

Many information retrieval problems can framed in the following way: given a set D of n documents and some query q, form a sequence of documents documents $d_1, ..., d_\ell$, where the rank in this subset corresponds to relevance to q. In order to evaluate the retrieval mechanism, we often must construct the ground-truth set of documents that should have been returned in response to the query, called the *relevant set* and denoted here $R_q \subseteq D$; we can then define a boolean relevance variable r_i by

$$r_i = \begin{cases} 1 & \text{if } d_i \in R_q \\ 0 & \text{if } d_i \notin R_q \end{cases} \tag{9}$$

A common question in information retrieval problems is the number of results to return. In general, the more results that are returned, the more the algorithm has a chance to return the results that are actually useful in answering the query; however, returning more results also increases the chances that an irrelevant result will be returned. We thus often fix a number k of results to be returned, and look at performance measures of the algorithm as we vary k. Two basic performance measures are *precision* and *recall.*

The first question we might ask is, "Of the first k documents in the ranked result, how many are relevant?" This quantity is called the *precision at rank k* of the result, and is expressed

$$\text{precision}(k) = \frac{1}{k} \sum_{i=1}^{k} r_i. \tag{10}$$

We can also ask the dual question: "Of all the relevant documents, how many are present in the first k documents in the ranked result?" This quantity is the *recall at rank k*:

$$\text{recall}(k) = \frac{1}{|R_q|} \sum_{i=1}^{k} r_i. \tag{11}$$

Note that these measures trade off between each other; a query that returns everything would have high recall and low precision, generally speaking. It is sometimes useful to combine these notions into an *average precision*, which is the average of precision across all ranks for a particular query:

$$\text{AveP}(q) = \frac{1}{|R_q|} \sum_{k=1}^{\ell} r_k \cdot \text{precision}(k). \tag{12}$$

TABLE 2.1. An illustration of a confusion matrix, showing the common interpretation of the four positions.

		Predicted	
		Negative	Positive
Actual	Negative	**TN**	**FP**
	Positive	**FN**	**TP**

This measure combines notions of precision and recall. It is highest when the ranking places all relevant documents before any irrelevant documents.

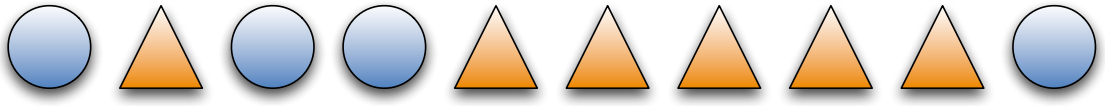
Note that all of the above measures measure performance relative to a single query only. When evaluating the performance of a retrieval algorithm, it is sometimes useful to aggregate a performance measure across an entire set of queries for the algorithm. This can be done with the Mean Average Precision (MAP), which averages the average precisions over each query. For some set of queries Q, the mean average precision is given by:

$$\text{MAP}(Q) = \frac{1}{|Q|} \sum_{q \in Q} \text{AveP}(q). \tag{13}$$

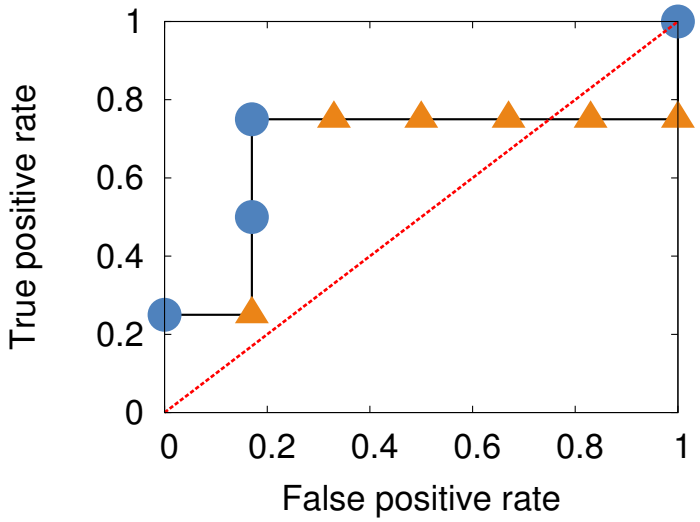
The MAP is a combined notion of precision and recall at all ranks over all the queries for which the algorithm is to be evaluated. Of course, there are many others.

2.5.2. Confusion matrices

Another tool for analyzing the accuracy of an information retrieval algorithm (or machine learning algorithm) is the *confusion matrix*. This is a 2×2 matrix containing four numbers: the number of *true positives* (returned results that are relevant), *false positives* (returned results that are *not* relevant), *true negatives* (non-returned results that are not relevant), and *false negatives* (non-returned results that *are* relevant). Note that the better the performance of the algorithm, the more of the mass of the matrix will be on the main anti-diagonal, rather than the main diagonal. Table 2.1 shows an illustration of a confusion matrix, giving the position of the values.



(a) A sample set of items. Blue circles represent relevant items, with orange triangles representing irrelevant items.



(b) The ROC curve for a retrieval algorithm that retrieves items in the order shown above. The dashed red line on the main diagonal represents performance at chance levels.

FIGURE 2.7. Example ranking of a set of items (a) with associated ROC curve (b). The ROC curve suggests the best performance is for $k = 4$.

2.5.3. ROC curves and AUC

There are other ways to get an overall sense for an algorithm's performance as the number of returned results, k, is varied. One of these is the so-called *receiver-operating characteristic curve*, or *ROC curve.* This analytical technique was first developed as an aid to tuning radar detectors during World War II **[GS66]**; it has since been introduced into many other areas. The ROC curve is a parametric plot of the true positive rate vs. the false positive rate for a retrieval algorithm as the number of returned results (k) is varied. As an example, suppose I have a retrieval algorithm meant to retrieve blue circles from a collection containing both blue circles and irrelevant orange triangles. Suppose that this retrieval algorithm ranks the 10 items in my collection as shown in Figure 7(a).

Thus for $k = 1$, the algorithm will return one relevant result, but will miss three, for a true positive rate of $1/4$ and a false positive rate of 0. For $k = 2$, the algorithm will return one relevant result and one irrelevant result, while still missing three relevant results, for a false positive rate of $1/4$ and a false negative rate of $1/6$. Plotting these rates for $1 \leq k \leq 10$ yields the ROC curve shown in Figure 7(b). Note that in general, an ROC curve which differs significantly from the main diagonal represents performance significantly different from chance. Optimal performance would appear as an ROC curve that follows the Y-axis to 1, then goes from $(0, 1)$ to $(1, 1)$.

We can also measure the performance of an algorithm by the integral of the ROC curve; this is referred to simply as the 'Area Under the ROC Curve,' or the *AUC*. A perfect retrieval algorithm will have an AUC of 1; one that returns all irrelevant results before any relevant result, being a perfectly terrible algorithm, will have an AUC of 0. An algorithm that performs at chance levels will have an AUC of 0.5. Note that an algorithm with an AUC of 0 is perfect in the sense that it is perfectly wrong; it can be transformed into a perfect algorithm by reversing the ordering it uses, so that it returns all relevant results *first* instead of last.

2.5.4. Kolmogorov-Smirnov statistic

The form of the Kolmogorov-Smirnov (KS) statistic that I later use is its *one-sample* form, which quantifies the similarity between a sample and a reference probability distribution. This tool is useful for determining if some empirical data was likely to be drawn from a specific distribution. It does not depend on the reference distribution, making it useful for cases where no more specific tools exist. Its weakness is that it requires a large number of data points to reach any useful levels of confidence.

The KS statistic works by comparing the empirical distribution function (EDF) of the sample (defined below) with the cumulative distribution function (CDF) of the given probability distribution. The null hypothesis is that the sample is drawn from the distribution in question; if the difference is too great, this can be rejected. The KS statistic's need for large amounts of data is not often a problem when dealing with Web data consisting of millions of points.

The KS statistic for a collection of n sampled values $X_1, ..., X_n$ and the cumulative distribution function $F(x)$ of the proposed probability distribution may be computed as follows. First compute

the *empirical distribution function* (EDF) of the observed values, given by

$$\text{EDF}(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x},$$

where the *indicator variable* $I_{X_i \leq x}$ is 1 when $X_i \leq x$ and 0 otherwise. Thus, the value of the EDF at x is the fraction of the observed values that are no greater than x, making it the empirical equivalent of the CDF. The KS statistic for this EDF and the cumulative distribution function $F(x)$ of the proposed probability distribution is then given by

$$D_n = \max_x |\text{EDF}(x) - F(x)| .$$

If the values X_i are indeed sampled from the probability distribution F, this value will converge to 0 as the number of input data points (n) rises. However, if D_n is large (its maximum value is 1, so 'large' in this case might mean greater than 0.2), we can conclude that there is a point of significant difference between the empirical distribution and the theoretical one. Thus, the null hypothesis (that the distributions are the same) can be rejected.

2.6. Analytical techniques

In this section I overview some general analytical techniques that are very useful in studying large social and information networks.

2.6.1. Plotting techniques

In 2.1.4 I promised graphing techniques to display 'heavy-tailed' distributions, such as those often seen for degree and strength, in a more useful manner. One of these techniques is that of plotting some distributions on *log scales*; the other is *log binning*. Each of these addresses the fact that when plotting a distribution that spans several orders of magnitude, we often want to focus on the behavior at those orders of magnitude rather than treating each bin of the histogram equally. For instance, we may care less about the difference between the bins for the values 1001 and 1002 than we care about the bins for 1 and 2. Log binning the data also has the useful property that it removes noise from the data, making general properties (such as the slope of the curve) more apparent. Another technique to this same end is plotting the *cumulative* distribution function of the data, possibly combined with plotting on a log scale.

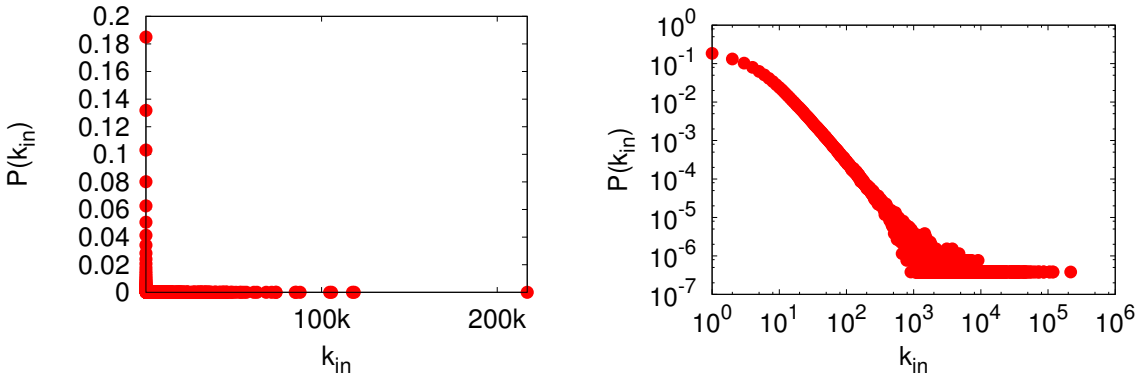


FIGURE 2.8. Two views of the same data, namely the distribution of in-degree for Wikipedia pages. The plot on the left has each axis on a linear scale; the plot on the right uses log scaling.

2.6.1.1. Log-scale plotting. One technique for effectively visualizing broad distributions is to plot the graph with one or both of the axes on a logarithmic scale, rather than a linear scale. This can help to visualize a distribution across all of its size resolutions. Figure 2.8 shows two views of the distribution of in-degree for English Wikipedia pages — one on a linear axis, and one on a logarithmic axis. The plot on the logarithmic axis makes this distribution much easier to see.

2.6.1.2. Log binning. Note that while plotting a distribution on a log scale compresses the display of the histogram bins, it does not actually reduce the number of those bins. Thus the space between 10^0 and 10^1 on the plot is represented by 9 bins, while the space between 10^4 and 10^5 is represented by 90,000 bins — even though both of these intervals have the same display space on the plot. I thus often make use of another technique called *logarithmic binning* or *log binning*, in which the bins themselves are on a logarithmic scale so that the size of the bins is no longer uniform. Of course, this requires normalizing each bin by its own size, as well as the size of the distribution, so that we preserve the necessary property for a PDF that the area under its curve be 1. The left plot in Figure 2.9 shows the same distribution of indegree, using both log and linear binning. The logarithmic binning is able to show much more resolution, especially in the tail of the distribution.

2.6.1.3. Complementary cumulative distributions. Another technique for smoothly visualizing the probability distribution of long-tailed variables is that of plotting the *complementary cumulative* distribution function (often confusingly also called the CDF in this context). While in other domains the CDF of a random variable X is taken to be $\mathrm{P}(x \leq X)$, the *complementary* CDF is defined by

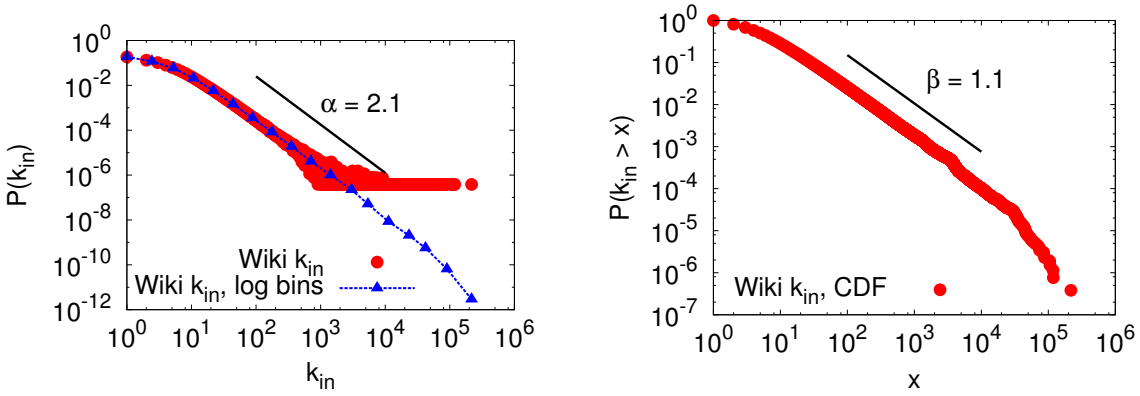


FIGURE 2.9. Two views of the indegree distribution of Wikipedia pages. The left plot shows the raw probability distribution histogram, both with fixed-width and log-width bins. Note that the log-width bins are able to show much more resolution in the tail of the distribution. The power-law guide to the eye shows the slope of a power law distribution with $\alpha \approx 2.1$. The plot on the right shows the cumulative distribution function of the same data. This has a similar smoothing effect, with a related exponent — $\beta = \alpha - 1$.

$P(x \geq X)$. The plot of this latter function has the following nice property: if X is a random variable distributed according to a power law with exponent α (corresponding to a linear curve with slope $-\alpha$), the slope of the curve of X's CDF will be equal to $\beta = 1 - \alpha$, corresponding to a power law with exponent $-\beta = \alpha - 1$. The right plot in Figure 2.9 illustrates this. Another name for the complementary CDF is the *exceedance*.

2.6.2. Heavy-tailed distributions

In a few previous sections I have mentioned the concept of a *heavy tailed* distribution. Intuitively, this is a distribution for which there is no clear cutoff beyond which larger values are very unlikely, or one for which that cutoff is very large. This is in contrast with a *peaked* distribution, of which the standard normal distribution is an example. Increasing the number of sampled items taken from a normal distribution is not likely to broaden the distribution, but increasing the sample size of items

taken from a heavy-tailed distribution likely will. The distributions of many values associated with real-world networks, such as degree and strength, exhibit heavy tails. The concept of a heavy-tailed distribution has a precise meaning in probability theory; a distribution of a random variable X is heavy tailed if

$$\forall \lambda > 0 \qquad \lim_{x \to \infty} e^{\lambda x} \Pr(X > x) = \infty \tag{14}$$

I never need this formal definition, relying on the intuitive one.

A commonly-cited example of a heavy-tailed distribution is the so-called *power-law distribution*. A random variable X is said to obey a power-law distribution when its probability distribution is

$$\Pr(X = x) \propto x^{-\alpha} \tag{15}$$

for some constant exponent α called the *scale factor* or *scaling exponent*. In real world networks, we often have $2 < \alpha < 3$. Due to the fact that real networks have finite size, power-law distributions reflecting real-world quantities always have a cutoff determined by the size of the systems. It is sometimes a source of disagreement whether a particular set of data fits a power-law, or if the data is really representative of a stretched exponential or another such broad distribution. Thus the more general terms 'broad distribution' or 'heavy-tailed distribution' may be used when the actual distribution is not important.

A commonly-mentioned power-law distribution is the Zipf distribution. This is named after the linguist George Zipf, who noticed that in natural language, a word's frequency is inversely proportional to its position in the ranked list of all word frequencies [**Zip49**]. The PMF of a Zipf distribution is given by

$$p(x) = \frac{x^{-a}}{\zeta(a)}, \tag{16}$$

where the constant a is the scaling factor, and ζ is the Riemann zeta function. Figure 2.10 contains the distribution of 100,000 numbers chosen at random from a Zipf distribution with $a = 2$, plotted on a log-log scale. Both the linear bins and log bins are shown. Note that the log binned version of the histogram describes a straight line on the log-log plot; this is characteristic of power-law distributions. The slope of the line is the exponent of the power law.

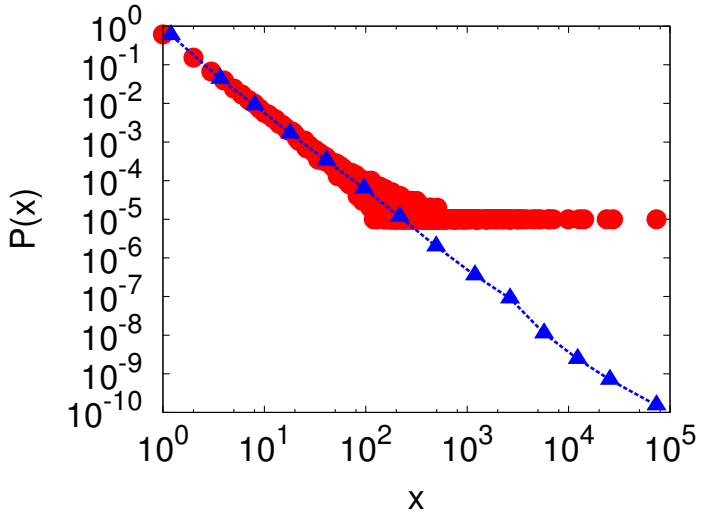


FIGURE 2.10. Distribution of the frequency of 100,000 numbers chosen at random from a Zipf distribution with parameter $a = 2$. Note the straight-line appearance with slope -2, corresponding to the power law exponent.

2.6.3. Maximum-likelihood fits of power-law data

Given a set of data and a guess about the possible distribution that the data might have, it is often useful to estimate the parameters that fit the distribution best to the data. In the case of simple distributions, such as the normal distribution, this is trivial. However, care must be taken when dealing with potentially power-law distributed data, in order to avoid fitting to a power-law data that would be better fit to a narrower distribution (such as a log-normal). The commonly accepted method for fitting a power-law distribution to data is that described by Clauset *et al.* [**CSN07**]. This paper also gives a very detailed treatment of power-law distributions in general, putting them in contrast with several other types of broad distributions.

In short, the method for fitting power-law distributions is divided into two steps: finding the lower bound, $x_{\min}$, on the power-law behavior, and finding the scaling constant α for the power-law behavior above $x_{\min}$. As an example, I apply the methods in the paper to the data sampled from the Zipf distribution in Figure 2.10; these data, along with the best-fit power-law alpha determined by maximum-likelihood approximation, are shown in Figure 2.11. Here, $x_{\min} = 1$.

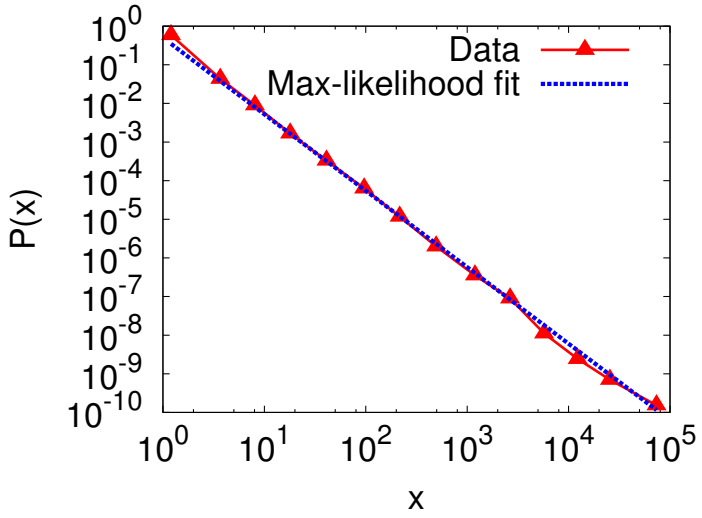


FIGURE 2.11. Zipf-distributed random data from Figure 2.10, with its maximum-likelihood power-law fit. The scaling parameter of the fit is $\hat{\alpha} \approx 1.98$ by maximum likelihood, very close to the true value of 2.0.

2.6.4. Other similarity measures

In this section I briefly review some general similarity measures that appear in this work in several contexts.

2.6.4.1. Jaccard. The *Jaccard coefficient* is a measure of the relative overlap of two sets. For two sets A and B, the Jaccard coefficient is defined by:

$$\text{jac}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{17}$$

that is, the size of the sets' intersection divided by the size of their union. This measure varies from 0, when the sets have no elements in common, to 1, when they are identical.

2.6.4.2. Pearson's r. More properly called *Pearson's product moment correlation*, this is a statistical tool for quantifying the linear dependence between two variables. Given values $X_1, \ldots, X_n, Y_1, \ldots, Y_n$ sampled from the random variables X and Y, respectively, we define the sample correlation coefficient r by

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2} \cdot \sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}, \tag{18}$$

where $\overline{X}, \overline{Y}$ are the sample means. This measure varies from -1 (perfect anticorrelation) to 1 (perfect correlation), being 0 when no correlation is present.

CHAPTER 3

RELATED WORK

In this chapter, I discuss work related to the problems attacked in this dissertation. These are roughly divided among the topics of *online popularity*, *graph growth models*, *meme tracking*, and *political discourse*, the latter being relevant to several case studies. Graph growth models are relevant in that they are essentially models of the growth of a certain type of popularity — that of the number of incoming links to a node or Web page.

3.1. Online popularity

In the discussion of Web popularity I include studies of the structure of the Web at large, as the in-degree of a page is an important measure of its popularity. An initial such study exposed a 'bow-tie' structure of pages, consisting of three major disjoint sets: a central strongly connected component (the *core*), a set of pages that can reach the core but that are not reachable from it, and a set of pages that are reachable from it but cannot reach it [**BKM+00**]. This structure is illustrated in Figure 3.1. Note the 'tendrils' that are connected to the 'in' and 'out' portions of the 'bow tie,' but cannot reach the core.

Several studies have used crawl data to analyze the temporal evolution of the Web, focusing on creation and destruction of pages, links, and the frequency and amount of change in page

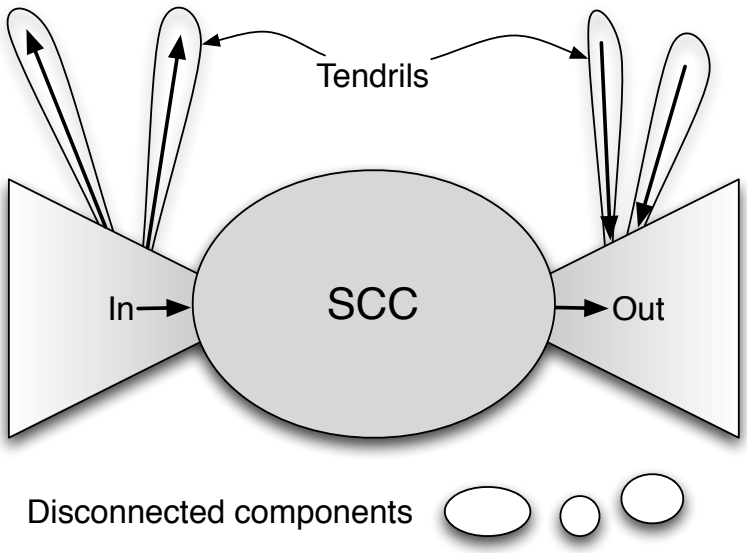


FIGURE 3.1. The 'bow tie' structure of the web, with the strongly-connected component in the middle [**BKM$^+$00**].

content [**BC00, FMNW03, NCO04**]. This approach, however, does not allow one to track individual pages or sites longitudinally in order to accurately monitor their popularity over time. Kleinberg [**Kle02**] studied the bursts associated with identifiable events in streams, such as the occurrence of a key phrase in a news feed. This approach allows one to detect hot topics as temporal bursts in word usage. Kumar *et al.* [**KNRT03**] expanded this notion to analyze the evolution of bursty communities in blogs. They also developed the concept of time graphs, which is similar to the methodology used here for tracking temporal patterns of popularity. On the modeling side, Barabasi [**Bar05**] suggests prioritization as one mechanism leading to bursts of activity. Mathioudakis *et al.* [**MKM10**] develop a model for attention in social media. Users are viewed as producers of information streams, made of units that may be noticed by other users. This model characterizes items such as blog posts by their 'interaction weights,' a proxy for the degree to which users noticing the items.

An initial issue facing a study of the sort presented later in this dissertation (Chapter 6) is the identification of a suitable popularity measure. In recent years, the mapping of large, complex information networks [**AJB99, BKM$^+$00, SMB$^+$07**] has led to identifying the number of links pointing to a node (its *indegree*) as a proxy of popularity in many domains [**SP06**]. In 3.2 I describe

several models of the evolution of indegree. The evidence that many social, technological, and information networks are characterized by stable heavy-tailed distribution of indegree pointed to a strong heterogeneity in the popularity and triggered the formulation of models aimed at explaining the emergence of such broad distributions using rich-get-richer mechanisms [**Sim55**] based exclusively on topology [**dSP76, BA99, KKR$^+$99**] or combined with content information [**Men04**]. While these models have the merit of introducing irreversible growth as an important element of network generation, the dynamics characterizing these rapidly changing systems have been seldom studied because to date it has been *infeasible* to observe the actual growth of an online network. The datasets we utilize, however, contain longitudinal information that makes it possible to observe their growth. Further we have access to traffic data, which we consider a more direct proxy to popularity as it represents human attention more immediately.

Web traffic is a proxy for online popularity. While the static properties of Web traffic have been fairly well investigated (e.g., its distribution across all pages or hosts in a given period of time [**MMF$^+$08**]), much less is known about how the traffic toward individual pages changes over time and what factors affect its dynamics, especially when this traffic is characterized by non-regular and intermittent activity.

Some prior work on the topic of popularity dynamics has focused on news. Wu and Huberman [**WH07**] performed a large-scale study of the news sharing site `Digg.com`, where users can promote links to articles they like by voting for them. This study tracked the total number of votes that each story receives through its lifetime, finding that this quantity follows a lognormal distribution. They further examined the decay rate of incoming votes for a story, providing insight into the onset and decay of a story's popularity. In general, the dynamics of short-lived events such as the news cycle are relatively well understood; popularity of individual items tends to be distributed according to a lognormal, and stops being accreted after around 36 hours in normal cases [**DAL$^+$06**]. When we broaden our range in considering any online Web page or topic, the distributions of the popularity measures we study on the Web and Wikipedia — node indegree in both systems, and traffic in the latter — fit a power law much better than a lognormal (6.3.1). Therefore the behavior of online popularity cannot in general be characterized by that of news-driven events. One possible reason for this is illustrated by considering the difference between the news story "Barack Obama

inaugurated as U.S. President," and the Wikipedia article on "Barack Obama." The latter's popularity subsumes that of the former, and of potentially many other news stories. While attention for a particular news item is short lived almost by definition, the popularity of a Web or Wikipedia page may be influenced by many news events over an indefinite time span.

The popularity of videos on the YouTube video sharing site has been studied by Szabo and Huberman [**SH08**] and Crane and Sornette [**CS08**]. These dynamics are found to be similar to those of news, but with different popularity classes depending on whether a video has been featured on the front page of the site, or is the type that is likely to be spread by social networks (a so-called *viral* video).

Compared to the existing literature about features of popularity trends such as those mentioned above, work on the potential causes for these trends is scarce. It has been shown that when users have access to popularity rankings (e.g. YouTube views or presence of a book on the New York Times bestseller list), they are more likely to disproportionately favor popular items [**DAL$^+$06, CS08, SDW06**].

Other recent work on human activity in the Web at large has focused on search engines [**FFMV06**] and Web traffic [**MMV05**]. In the latter study, Meiss *et al.* find that the distribution of the traffic directed at hosts on the internet is very broad, well fit by a power law with exponent less than 2. For such a broad distribution, the mean is not a meaningful quantity. Thus, it is not meaningful to consider the "average" popularity of a Web host. These findings were confirmed in a later study on a different data set [**MMV11**].

3.2. Graph growth models

Focusing on indegree as a popularity measure, several models have been proposed to interpret the evolution of this quantity. The best known network growth model is preferential attachment [**BA99**], addressed later. It is the foremost example of a class of *rich-get-richer* growth algorithms, in which a node's probability to acquire new links (or popularity) is an increasing function of that individual's current number of links (popularity). The following are a number of network growth models, including several examples of the rich-get-richer class. The following models all seek, indirectly or indirectly, to produce graphs with (at least) the following two properties: a small *diameter* and a high *clustering coefficient* [**WS98**]. The diameter of a graph is the distance between the

most-distant pair of nodes, and can be computed in a directed or undirected fashion. The clustering coefficient, intuitively, is high when nodes share many mutual neighbors. Both of these properties are true of many real-world networks. Graphs with a low diameter (relative to the number of nodes) and a high clustering coefficient are referred to as *small world* graphs.

3.2.1. Watts & Strogatz

One of the first network evolution models was that proposed by Watts and Strogatz [**Wat99**]. This model is not truly a growth model, since it does not iteratively add vertices but rather starts with the final number of vertices already present; still, I include it for completeness. The model is initialized with a regular lattice of N vertices, where N is also the desired final number of vertices in the graph. This lattice, most frequently one-dimensional though this is not a requirement, is such that each vertex is initially connected to its k nearest neighbors to form a ring. The model then chooses a fraction p of the edges at random, and rewires one endpoint of each to another node, also chosen uniformly at random (modifications to the model have proposed rewiring both ends of the edge, as well as allowing self-links and duplicate links, for ease of analysis.) The initial lattice causes the clustering coefficient to be high, as initially the edge sets of neighboring nodes overlap almost entirely. The random rewiring adds the possibility for long-range hops, bringing the diameter of the network down.

This model is unsuitable for the study of the Web, and most other large networks, for several reasons. Its degree distribution does not match any commonly found in the real world (it is Poissonian); further, it is hard to argue that the process it describes models any found in the real world. As noted earlier, it does not permit the addition of new nodes once the process has started, which is certainly not the case in many real networks.

3.2.2. Barabási & Albert

This model [**BA99**] had immense influence, and it was its creators that coined the term *preferential attachment*, commonly used to describe the "rich get richer" behavior exhibited by link attachment in growing graphs. The model begins with an initial random network (for instance, generated by the Erdős-Rényi model), to which vertices are iteratively added. Each new vertex links to m existing vertices, with the probability of linking to a particular vertex j given by the fraction of

all previously-existing links that already connect to j. This model has the advantage of producing a graph with a power-law degree distribution; it further can be argued to resemble somewhat the growth of real-world graphs. However, this model is also not completely realistic; it produces undirected and acyclic graphs in its simplest form. In its extension to directed networks, it produces graphs in which out-degree is constant. Many real-world graphs have none of these properties. Further, this model requires global knowledge of the network in assigning a new node's edges, something that the author of a Web page or Wikipedia article (for example) is unlikely to possess. Dorogovtsev *et al.* [**DMS00**] have developed extensions to this model that address some of these concerns (for example, producing directed graphs with unspecified outdegree distributions, and allowing the slope γ of the power-law fit on indegree distribution to be modified). Still, while this family of models may adequately describe the state of a graph at any fixed point in time, it cannot model the dynamic processes by which the graph arrived at that state.

3.2.3. Heuristically Optimized Tradeoffs

This model, due to Fabrikant *et al.* [**FKP02**] begins with all the nodes in the nascent graph represented by points distributed uniformly at random over a unit square. It then builds trees of connections reaching out from each node, where nodes may choose to connect either to nodes which are nearby in the geometrical space (perhaps within a "cone of influence" emanating out from the node), or farther away but more popular (i.e., with greater indegree). Thus, the distribution of the nodes in a geometric space can be seen to model the distribution of web pages in the space of lexical similarity, or any other similarity space that might be imagined. With proper parameters trading off between local connectivity and popularity, the HOT model can yield trees with power-law degree distributions.

3.2.4. Traffic-Driven Growth

This model is similar to the preferential attachment (PA) model of Barabási & Albert previously described; however, it attempts to consider traffic when assigning new links, rather than using existing link structure [**BBV04**]. In this model, we begin with an initial small collection of nodes connected by weighted edges, where the weights are taken to represent the amount of traffic that is flowing over each edge. New nodes are then iteratively added to the graph, one at each timestep;

a new node then assigns m links to existing nodes so that the probability that an existing node receives a link is proportional to its total in-strength, where m is a parameter. New edges are given weights, and existing weights are adjusted to reflect the passage of traffic along the newly-added edges. We know from later work [**MMF$^+$08**] that inlinks do not accurately reflect the popularity of sites (in terms of the number of visitors) in many cases, so this model may have an advantage in realism over the PA model for this reason.

3.2.5. Copying

This model, proposed by Kleinberg *et al.* [**Kle99**], posits that the cloning of existing nodes (and their edges) is a driving force behind network growth. Under this model, a graph grows by iteratively adding vertices. A uniformly random number m of new edges is chosen for each new vertex. With a certain probability p (given as a parameter), these m edges are attached randomly to other vertices; however, with probability $1 - p$, edges are copied directly from a randomly-chosen existing vertex. If that vertex has fewer than m links, another is chosen until m links in total have been copied. This model does produce graphs that have power-law degree distributions, and does not necessarily require global knowledge of the state of the entire graph. This model can be thought of as a local approximation of preferential attachment.

3.2.6. Growth by Content

The models presented thus far are able to reproduce the degree distributions of many types of real-world networks. However, when applied to the Web (and other types of document networks), there are other statistics that might be reproduced, such as the distribution of content similarity among documents connected by hyperlinks. A model that grows graphs by content similarity as well as preferential attachment, proposed by Menczer [**Men04**], addresses this issue. In this model, pages are added iteratively to a growing graph as in many other models; however, links are formed to existing pages by either choosing to link to a page which is similar in content, or choosing to link to a prestigious page regardless of content similarity. The graph produced by this means has both a realistic degree distribution as well as a realistic distribution of linked page content similarity. The probability with which new pages link to similar vs. prestigious pages can be tuned in order to match the content similarity distribution of various corpora.

3.2.7. Growth by Ranking

This model, due to Fortunato *et al.* [**FFM06**], dispenses with the fixed notion of indegree as the prestige that attracts incoming links, and allows prestige to be assigned by an arbitrary function. The model works by iteratively adding nodes to a *ranked list* of existing nodes. Each new node assigns m links to existing nodes, where the probability that a new node links to an existing node j is a function of j's position in the ranking. More precisely, for a parameter $\gamma > 0$, the probability $\mathrm{P}(j)$ of linking to existing node j is given by

$$\mathrm{P}(j) = \frac{R_j^{-\gamma}}{\sum_k R_k^{-\gamma}}, \tag{19}$$

where R_j is the rank of node j. The model is quite robust to the choice of the ranking function; indegree (or age) is a possibility, but the authors show that any arbitrary static ranking produces a power-law degree distribution in the resulting graph. The exponent α of this resulting power-law is related to the parameter γ by

$$\alpha = 1 + 1/\gamma. \tag{20}$$

Further, this model can be adapted to remove the requirement for global knowledge of the existing rankings when assigning a new vertex's links. This is done by restricting each new node to a subset of the existing nodes, from which it must choose its outlinks. The link probability then depends on the existing nodes' ranking within this subset. The authors show that as long as the size of this local subset is not too small (relative to the total number of nodes) the model still produces graphs with power-law degree distributions.

3.2.8. Forest-fire model

The forest-fire model, introduced by Leskovec *et al.* [**LKF07**], is named for the way connections spread from a designated number of 'ignition points,' or *ambassadors.* The model works by the repeated addition of nodes, one per timestep. Each new node selects some number w of ambassador nodes, uniformly at random, and starts a 'fire' at each. This 'fire' burns from each chosen ambassador w_k, crossing the outlinks of w_k, to recursively set fire to the neighbors of w_k, with some *forward burning probability* p_f. The 'fire' may also burn backwards, across w_k's inlinks, with

a *backward burning probability* p_b. The 'fire' thus continues recursively 'burning' nodes and spreading to their neighbors until it extinguishes itself; when that happens, the originally added node is connected to all the nodes to which its ambassadors set fire.

The authors compare this process to that followed by an author exploring what papers to cite for her new paper; she might first pick some number of related publications, then reading their bibliographies to find new papers to cite.

3.2.9. Triangle-closing model

This model is based on the empirical observation by its authors that many new edges in a social network close triangles between two neighbors of a node [**LBKT08**]. That is, many new edges in a social network are between a person and a friend of a friend of that person. The model implemented to take advantage of this assumption concerns itself only with the addition of edges to an already-existent graph; the arrival of new nodes must then be modeled separately by another mechanism. At each timestep, a new edge arrives at a particular node u (chosen by some method outside this model). The node u chooses one of its neighbors v by some ranking method f; it then chooses one of v's neighbors w by some other (possibly different) ranking method g. It then forms the new edge (u, w). The authors explore several ranking methods f, g for choosing the neighbor and second-degree neighbor v and w, showing that choosing both nodes uniformly at random among all possible neighbors works remarkably well, performing not significantly worse than the more sophisticated measures. This model produces graphs with power-law distributions of degree, as well as realistic clustering coefficients. It also captures network degree better than does a baseline method based on preferential attachment. The model requires no global information, as each node only needs to know the friends of its friends in order to make an edge-attachment decision.

3.3. Memes and social media

While the popularity of web page can be measured by the number of visitors or in-links reaching it, the popularity of a *meme* or an *idea* is somewhat harder to quantify. This goes along with the fact that it is not always clear what constitutes an appropriate level of resolution in recognizing memes in the first place.

Much current research has been on identifying trends or memes in blogspace. Adar *et al.* [**AA05**] present a study of the propagation of a certain piece of information (in this case, a link to a Web site) through the blogspace, in effect tracking the passage of this information through the collective consciousness. Another paper by Leskovec *et al.* [**LAH06**] studies the propagation of recommendations among networks of friends, and is applied to viral marketing. Sinha & Pan explore popularity dynamics in a diverse set of areas, finding that the distribution of popularity among a number of choices often exhibits a log-normal or power-law behavior [**SP06**]. They further propose several models to explain these behaviors.

Social media have been characterized as a distributed sensor network that can tell us about the natural world. Sakaki *et al.* mine Twitter for earthquake-related utterances, using these to notify subscribed users of earthquakes in a manner they claim is faster than the official system for such notifications [**SOM10**]. The U.S. Geological Survey has undertaken a similar project [**Ear10**].

Asur and Huberman mine Twitter to predict the popularity — as measured by box office receipts — of newly-released movies [**AH10**]. They achieve results better than the gold standard for that industry. Galuba *et al.* predict the popularity — number of mentions — of a meme (encoded as a URL) on Twitter [**GAC$^+$10**]. They define a relatively narrow problem, that of predicting *which users* will propagate *which URLs*, when those users have previous seen that URL posted by one of their neighbors. They are successful in predicting more than half of the URL mentions in their data set, with less than a 15% false positive rate.

Work in epidemiology is also applicable to the study of memes, if a meme is viewed as a contagion that passes between humans in their communications with each other. Morris [**Mor00**] focuses on the topic of contagion in local-interaction systems. Pastor-Satorras and Vespignani [**PSV01**] examine the spread of a computer virus epidemic in a computer network, and Colizza *et al.* [**CBBV07**] consider the spread of epidemics over complex real-world networks in general. Finally, Lind *et al.* [**LdSAH07**] explore a model for the propagation of information based on gossip, in which nodes propagate a piece of information regarding a single "victim." The impact of large social systems, such as those we study, is explored in work by Tapscott and Williams [**TW06**]. This work focuses especially on the impact of these systems, which enable collaboration on a large scale by a distributed group of experts using the Internet, to business and economics.

CHAPTER 4

DATASETS

This chapter details the data sets that I use for experiments discussed in later chapters.

4.1. Wikipedia pages

The Wikipedia is a large collaborative online encyclopedia, available online at `http://www.wikipedia.org`. Its largest 'edition' is in English, containing millions of unique pages, edited by hundreds of thousands of registered users and an unknown number of anonymous users. Every edit to the Wikipedia is tracked, and all of these previous edits are available for study as well, up to about March 2007 (at which point the size of the dataset prohibited the Wikimedia foundation from continuing to make the full edit history available). This allows the reconstruction of the Wikipedia as it was at any previous point in time, enabling longitudinal study at an arbitrary time resolution.

Other authors have studied various instances of the network of Wikipedia pages and links (although not longitudinally), finding it to have the important properties of the Webgraph at large [**CSC+06, ZBSD06**]. Further, I know of two previous longitudinal studies of the Wikipedia. In the first [**BCD+06**], the authors consider a snapshot of the English Wikipedia taken every three months, for 17 snapshots in all (they do not make use of the full edit history). They examine the growth dynamics of the number of articles, updates to articles, visitors to pages, and registered editors, and find that these all seem to be growing exponentially. They also find that the indegree

distribution stabilizes very early, in the sense that it displays a power-law decay whose exponent has remained relatively unchanged for the last several years. The second study focuses heavily on the core activity of Wikipedia — that of individual editors modifying pages [**AMC07**]. The authors find that the process of accreting edits to an article is a self-similar process growing exponentially. In particular, this implies that articles may experience bursts of edits, and there is in general no expected starting time or duration of these bursts.

While it has been observed that the growth of the Wikipedia has slowed of late, my data refers to a time period in which the English Wikipedia as a whole was growing exponentially (e.g. in number of topics).

4.1.1. Data preparation

The Wikipedia dumps are in the form of compressed XML files which give the full text of every revision and the date and time at which it was made, as well as some details about the user who created it — their username if they are registered, part of their IP address if they are not (the full IP address is not provided, likely due to privacy concerns). I parse this data to produce a matrix, in which the rows represent pages and the columns represent dates. The entry for a particular page at a particular date represents the indegree of that page on that date.

Wikipedia contains a number of *redirect* pages — pages meant to bring a user to an article under a more correct spelling, or to the common definition of a term. For instance, the Wikipedia page for "Charles Dodgson" is a redirect to "Lewis Carroll," as it was by this pseudonym that he was better known. I handle these pages by rewiring links around them as shown in Figure 4.1. Wikipedia also contains a number of *special* pages, such as pages about particular users, or pages discussing proposed or controversial changes to articles. I omit these, focusing on the articles themselves.

Following this processing, the English-language Wikipedia as presented here consists of 3,293,102 vertices connected by 30,541,867 edges at the latest date available. This latest date is approximately March 2007, with historical data being available for the six previous years, since January 2001. Figure 4.2 shows the indegree distribution of the network, measured on January 1st in three successive years. Note the high degree of stability year to year, despite the exponential growth present in the system.

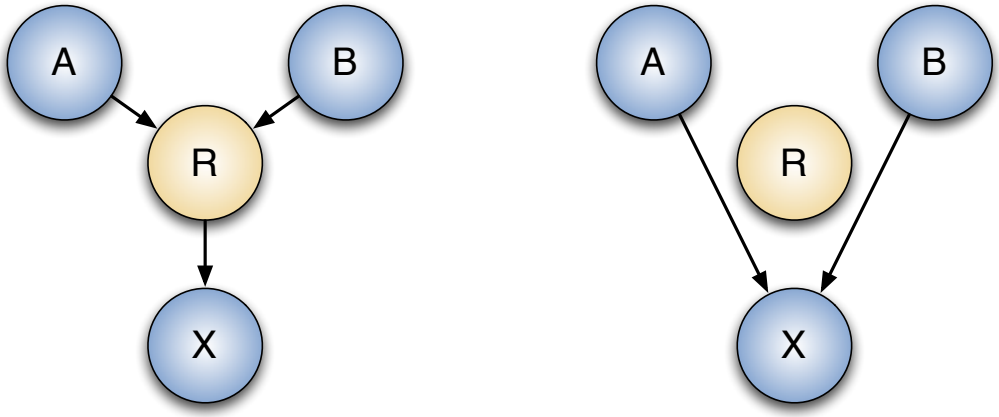


FIGURE 4.1. An example of rewiring links around a redirect page. The illustration on the left contains a redirect page R, linked two by pages A and B and pointing to a target page X. The illustration on the right shows the links from A and B rewired to point directly to the target page. The redirect page R is not removed from the graph, as it may become a non-redirect page in a subsequent time quantum.

4.1.2. Software systems

The English-language Wikipedia consists of over 1.4 terabytes of compressed XML, giving rise to some computational challenges in working with it efficiently — especially on the desktop-class machines available for this research at the time it was done. This section overviews the software systems I implemented to this end.

4.1.2.1. Data format. Wikipedia data is presented as an XML 'dump' of all revisions for each page in the dataset. Pages are presented in alphabetical order of their titles, with all revisions for one page being present in the stream before the next page — this might be termed a 'page-major' ordering. Figure 4.3 illustrates the organization of this stream. The size of the data file necessitated actually considering this data to be a stream over which only one pass was possible. A straightforward approach to parsing the XML with a so-called *DOM-based* parser, which requires building the entire document-object model (DOM) tree in memory, would have failed. Thus, I implemented a streaming (*SAX-based*) parser which only needs to keep a few revisions of the most recent page in memory at once. This parser, given a desired resolution (e.g. daily, weekly, or hourly), would parse the input XML and produce (among other things) a matrix containing the indegree of every page

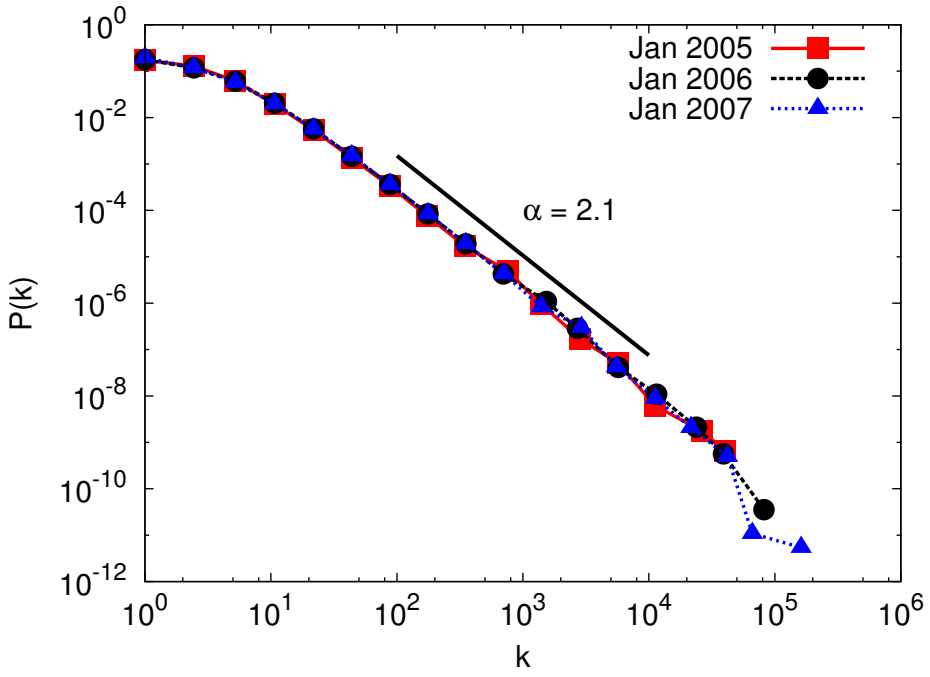


FIGURE 4.2. Distribution of indegree for the English Wikipedia as measured on January 1st in 2005, 2006, and 2007. Note that the distribution is very stable, though the size of the finite size cutoff grows larger as the system itself grows with time. The guide to the eye indicates the slope of a power law with exponent $\alpha = 2.1$.

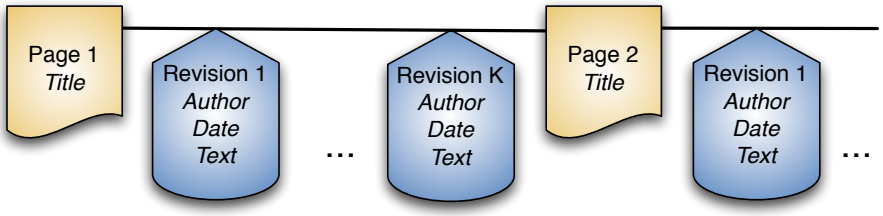


FIGURE 4.3. Organization of data in the Wikipedia dump XML file. Longitudinal data for each page was provided before data about any other page, and the size of the file necessitated that the parsing program have memory usage on the order of the number of revisions for any single page.

```
def process_next_object(obj):
    if obj.is_page():
        # Flush the last revisions of the last page,
        # and prepare for new page.
    else:
       # this must be a revision
       if obj.when > next_date_needed:
           # Then save the last revision for that date
           mark_effective_revision(last_revision, next_date_needed)
           next_date_needed = get_next_date_needed()
       else:
           # Don't do anything; there may be another revision
           # between this one and the next date we need.
           last_revision = obj
```

FIGURE 4.4. Pseudocode for the main stream-processing code. This code is called with each subsequent event from the stream. It tracks which revisions were active for each page at each date of interest.

at the beginning of every time quantum in the range for which there was data. The pseudocode for the main body of this parser is shown in Figure 4.4. The code here is called for each new object in the stream (either a new page or a new revision). It maintains a list of all the dates for which output is required, and tracks the 'active' revision for each page at each desired date — the state at which the page was. When it determines the active revision for a particular date, it parses the text of that revision to extract all its hyperlinks, as well as its raw text when all markup has been removed. These are stored in files named according to the date they represent, thus translating the page-major ordering of the input stream to a date-major ordering. A second pass over each of these sets of files then builds the indegree matrix for all pages, and an indexed collection of the TF vectors for each page at each timestep. The process for building the latter is straightforward, but the calculation of indegree bears more explanation. Note in particular that it was not possible to compute indegree in one pass, as maintaining global information for the in-links of all pages across all times required a prohibitive amount of memory.

The output of the previous process as relates to indegree is a file for each relevant date. This file contains an entry for every page that existed at that date, together with the names of all the pages linked to by that page. Finally, the file also noted if the page was, at the given date, a redirect page. Given this information, a second pass could associate with each page all the other pages that were linking to it at the date in question. It could also perform the re-wiring around redirect pages (as shown in Figure 4.1). Because every view of the data was too large to be stored in memory, an

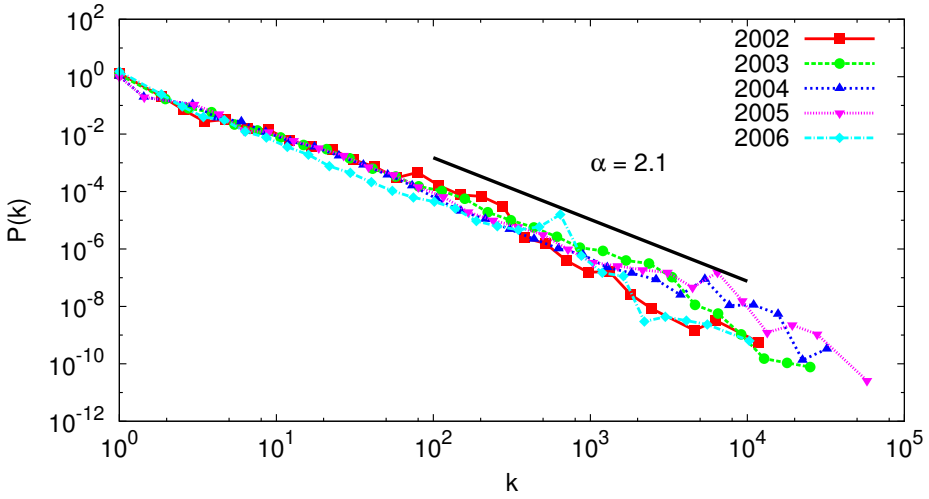


FIGURE 4.5. Distribution of indegree for each of the Chilean Web graphs. These distributions are not as stable as those for the Wikipedia, but still quite stable. The guide to the eye corresponds to the slope of a power-law distribution with exponent $\alpha = 2.1$.

efficient disk storage method was needed for the final form of the data — one that both consumed a reasonable amount of disk space and allowed for fast retrieval. I ruled out language-specific solutions for portability and efficiency reasons. Instead, I implemented a binary format consisting in most cases of two files: a data file that contains page identifiers and statistics, and an index file that stores the offset (within the data file) of where the data for each date began.

The Wikimedia project also releases dumps of the *current* state of the various editions of Wikipedia, without historical information. As such dumps are much smaller than those that contain revision information, they continue to be available for dates later than March 2007. Their format is just the same as for dumps with revision information, except that each page in the stream is followed by exactly one revision — the most recent one. This is convenient as it allows the above process to also work well for processing them.

The longitudinal data described above is used in analysis discussed in Chapter 6; data for a snapshot of the Wikipedia is used in analysis presented in Chapter 5.

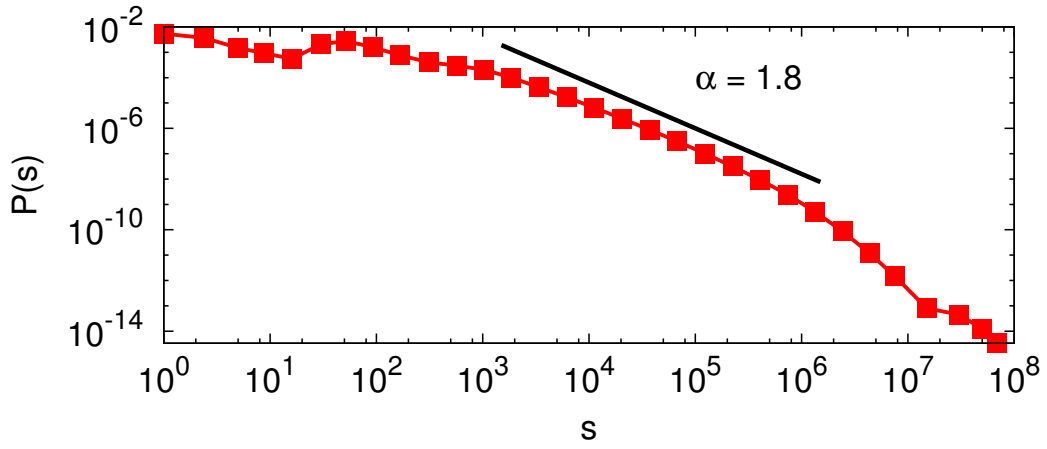


FIGURE 4.6. Distribution of the number of hits (popularity, or *strength*) received by individual Wikipedia topics, as measured at the Wikipedia proxy server, over one year. The guide to the eye indicates the slope of a power law with exponent $\alpha = 1.8$.

4.2. Chilean Web

This data, made available courtesy of the TODOCL search engine (`http://www.todocl.com`) consists of one crawl of the .cl top-level domain for each of the years 2002-2007. Perhaps not surprisingly, this data has the important properties also present in other samples of the Web [**BYP06**]. The only statistic available in this dataset is indegree; further, it is only available at yearly intervals. However, no work is required to extract it as it already present in the data set. Its format is documented in a technical report [**Cal99**]; it was necessary only to write a simple program to convert the indegree represented here to the same format used for the Wikipedia indegree. The largest graph in this dataset consists of 3,252,779 pages connected by 23,708,724 edges. Figure 4.5 shows the distribution of indegree for all these graphs.

This data is used in analysis presented in Chapter 6.

4.3. Wikipedia hits

This data set comes from D. Mituzas, a former software developer for the Wikipedia project who has been logging hits to the Wikipedia proxy server. It is available online at `dammit.lt/wikistats`; note that its availability is limited to the most recent few months. The data is formatted as compressed text files, one for each hour, containing record tuples of (*language code, article*

title, hit count). The data set was initially filtered to retain English Wikipedia pages by considering the language code. 'Special' pages (e.g. talk, image, and user pages) and pages that did not appear in Wikipedia as of June 2008 were filtered out. Data collection was initiated in February 2008 and continued until March 2010, although my analysis in this dissertation is restricted to the 13-month timespan between 1st September 2008 and 1st October 2009. For the purpose of this study, this data set has two shortcomings: first it does not contain referrer information, making it impossible to determine where (Wiki article or Web page) the visit to a page has originated from; second, it does not provides information on what type of agent generated a hit (human or crawler). We started collecting this data almost a year after the last date available in the Wikipedia full-history dataset. This makes it impossible to directly compare a page's in-strength with its indegree for the same time period. Figure 4.6 shows the distribution of the number of hits s received by each page, revealing the same broad features already observed for traffic to Web hosts [**MMF$^+$08, MMV11**]. I refer to this data set as 'page hits,' to distinguish it from my other source of traffic data, to be discussed next.

This data is used in analysis presented in Chapter 5 and Chapter 6.

4.4. Indiana University traffic data

This data set, due to Meiss [**MMF$^+$08**], is a log of Web requests outgoing from all of Indiana University. This data set includes records of the (anonymized) Web browsing activities of about 100,000 faculty, staff, and students of Indiana University from March 2008 to October 2009. The data consists of tuples of the form (*timestamp, agent type, http referrer, target host, target path*).

Further discussion of this dataset is reserved for the only chapter in which it is used, Chapter 5, where it referred to as the 'traffic' dataset.

4.5. Google trends data

Google publishes data about search trends it observes at `trends.google.com`. Given a query, this site will provide tuples of the form (*date, volume*) representing search trends for that query. The *date* given is at the resolution of weeks, and the *volume* represents the relative volume of queries in that week with respect to an average volume for that query. Rate limits restrict the number of queries that can be addressed to Google Trends. We collected Google Trends data for several

User vitals:

- Number of posts
- Number of followers
- Account creation date
- Screen name
- User description
- Real name
- Unique ID
- Latitude / longitude

Post vitals:

- Post date
- Post text
- Replied-to user
- Retweeted user
- Unique ID

FIGURE 4.7. A subset of the fields available in each post from the Twitter 'gardenhose.'

hundred Wikipedia topics in order to perform correlation with article hits data, as described in Chapter 5.

4.6. Twitter data

This is a corpus of posts from the popular microblogging site Twitter (`http://www.twitter.com`). The data I use is from the six-week period preceding the 2010 midterm congressional elections in the U.S; that is, from September 14th until November 1st, 2010. The number of posts, or *tweets* in Twitter parlance, observed in this time range was approximately 354.5 million.

These tweets are made available in real time by Twitter by a mechanism known as the *gardenhose*; they represent a small, though unknown, sample of the tweets being submitted to Twitter at approximately the same time. Thanks are due to Bruno Gonçalves for collecting the data and making it available [**GPV11**]. Figure 4.7 contains a list of a subset of the fields available in each tweet.

I use this data in Chapter 7 and Chapter 8. In each of these chapters I perform slightly different pre-processing and filtering on the data to focus on a particular problem. I describe these processes in those chapters.

4.6.1. Twitter terms

Twitter users have evolved a rich set of conventions within their 140-character length constraints. When a user means to refer to another user directly, she may do so by including the other user's Twitter screen name, prefixed by an '`@`' sign; I refer to such instances as 'mentions.' Twitter users can indicate the topic of their tweet by including in the tweet one or more tokens preceded by a hash sign (`#`). These tokens serve the same annotation purpose that, for example, tags serve on social bookmarking sites, and are referred to as *hashtags*. Some examples of hashtags are:

`#gop`: For marking discussion about the Republican party in U.S. politics.

`#obama`: For marking discussion about U.S. President Barack Obama.

`#twihards`: Used by self-considered 'die-hard fans' of the fiction series *Twilight*.

The number of these hashtags is vast, and the process by which the community settles on a particular tag for a particular subject is an interesting one (the previous rather obvious examples notwithstanding). A final relevant bit of Twitter post mark-up is the *retweet* indicator, represented by the characters `RT` followed by a mention for a particular user, as follows:

```
UserA: RT @UserB Check it out! http://some-url.com
```

This would indicate that User B posted the original 'check it out!' message first, and that User A is redistributing User B's message to his own followers while giving attribution credit to User B. This *re-tweeting* is a powerful means of rapid information spread on Twitter, and is the second way that users can interact directly with each other (along with mentions).

4.6.2. Diffusion networks

For any collection of tweets, we can build two distinct networks of users: one in which two users are connected if one has mentioned the other in a post, and another in which two users are connected if one has retweeted the other. Specifically, I connect two users A and B with the directed *mention* edge (A, B) if A mentions B in a post (causing the message containing the mention to appear in B's home screen). I connect two users A and B with a directed *retweet* edge (A, B) if B retweets A, since that retweet is an indication that information passed from user A to user B.

The units of information passed along in a retweeting cascade are called *memes*. The concept of a meme is a very broad one, referring generally to any idea which can be transmitted among

people. Here, however, I consider a meme to be one of four things: a hashtag, a URL, the username of a Twitter user, or the text of an entire tweet with all of the previous three markup items removed. When a network is built from all the tweets mentioning a particular meme, I call this the *diffusion network* of that meme. It is a network of all the user-to-user interactions involving that meme, where the directions of the edges involved can be thought of as approximating the direction of the information flow between two users — the direction along which the meme was transmitted from one user to the other.

I explore these diffusion networks in Chapter 7 in the context of political discourse, and look at some applications in Chapter 8.

There is a third user network, the declared social network of who follows whom in the system. In this dissertation I do not consider the declared social network, focusing instead on the dynamic diffusion networks of ideas and concepts (memes).

CHAPTER 5

ATTENTION IN ONLINE NETWORKS

As a first step towards understanding the dynamics of popularity online, I performed a number of relatively straightforward experiments. Popularity is ultimately the sum of human attention over a fixed span of time; understanding popularity can then be aided by understanding how the attention of the humans who drive it flits from topic to topic. This chapter mainly focuses on Wikipedia, and is divided into two endeavors — gaining a macroscopic understanding of where the Wikipedia is situated in relation to other sites on the Internet, and gaining a more microscopic understanding of how users move between pages inside the Wikipedia. I address each of these in turn.

Figure 5.1 contains an example of the kind of dynamics that are possible. Shown here are the number of page views, over time, of three pages from the English Wikipedia — the page for 'Biology,' and the pages for 'Barack Obama' and 'Michael Jackson.' The page for 'Biology' displays a predictable weekly cycle, with peaks around final exam weeks; the latter two are more bursty. The hit count for Barack Obama is dominated by spikes at the times of two major events — his election as U.S. president and his inauguration. The page for Michael Jackson shows a similar spike related to a news event, that of his untimely death.

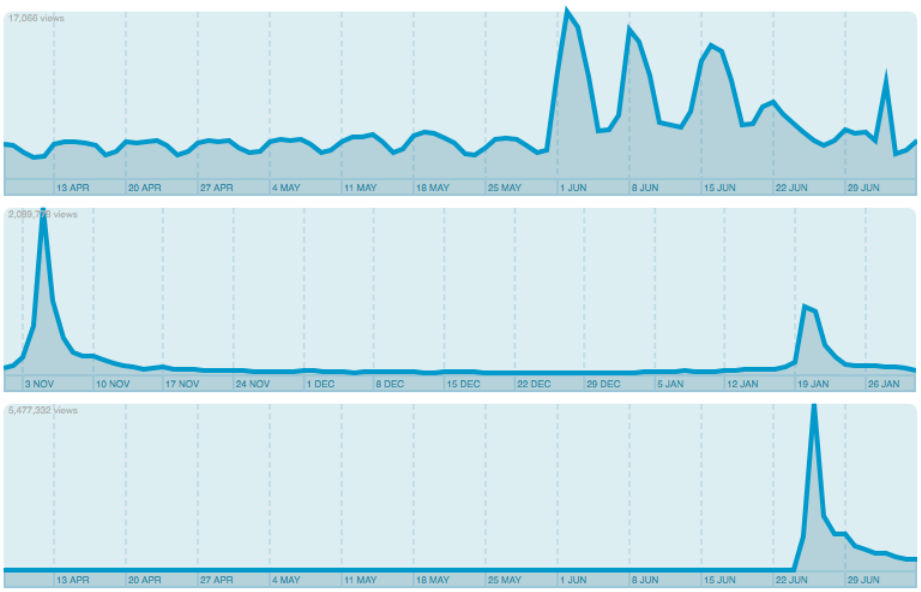


FIGURE 5.1. Comparison between the temporal traffic patterns of three different Wikipedia topics, visualized by `wikirank.com`. 'Biology' (top) displays a predictable weekly cycle, as well as peaks in demand around final exam weeks. 'Barack Obama' (center) and 'Michael Jackson' (bottom) are instead dominated by exogenous news events.

5.1. Macroscopic Properties

5.1.1. How Users Come and Go

My first analysis is aimed at understanding how users reach a Wikipedia page and to what extent their visit fulfills their informational needs, or leads to new resources linked from the page. This analysis is performed on the traffic data introduced in 4.4. Specifically, I build the weighted network induced by considering only tuples whose source or target pages are in the English Wikipedia. Figure 5.2 shows the degree and strength distributions of the resulting network.

When I consider the traffic data for which either the referring or target page is a Wikipedia article, I find that Wikipedia is a traffic sink: the volume of traffic originating from Wikipedia articles (either toward external pages or other articles) is about 30% less than the volume flowing into Wikipedia. Tables 5.1 and 5.2 show the 10 referring and target hosts for Wikipedia articles that account for the most traffic. The top 10 referring hosts account for 95% of incoming traffic.

TABLE 5.1. Top referring hosts for Wikipedia articles.

referring host	**share**
en.wikipedia.org	44.81%
google.com	33.99%
empty referrer	9.20%
wikipedia.org	3.56%
search.yahoo.com	1.57%
search.live.com	0.72%
bing.com	0.60%
stumbleupon.com	0.27%
search.msn.com	0.23%
ask.com	0.08%
total	**95.03%**

TABLE 5.2. Top hosts reached from Wikipedia articles.

target host	**share**
en.wikipedia.org	69.66%
indiana.edu	6.18%
boost.org	3.74%
dlib.indiana.edu	1.16%
kinseyinstitute.org	1.10%
omrf.ouhsc.edu	0.56%
banknoteworld.com	0.41%
imdb.com	0.37%
cs.indiana.edu	0.32%
jcmc.indiana.edu	0.23%
total	**83.73%**

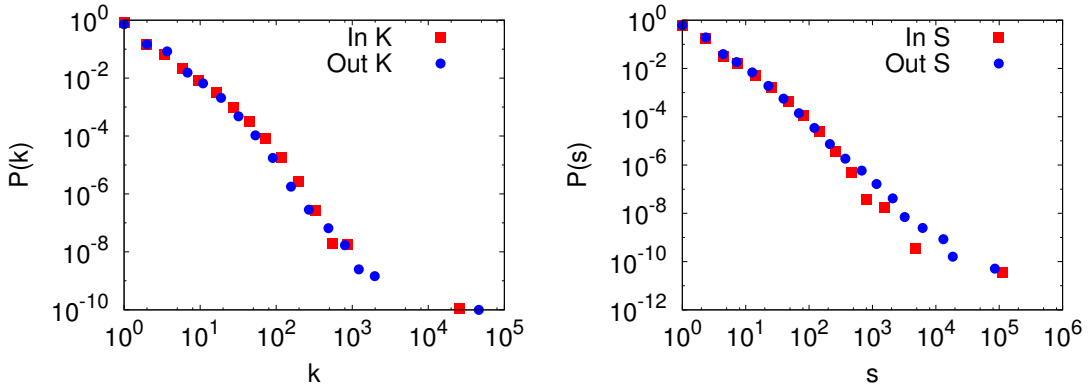


FIGURE 5.2. Degree (left) and strength (right) distributions for the network induced by traffic to and from Wikipedia.

Our data suggests that most users access articles either from other Wikipedia pages, or are directed there by search engines (mostly Google). About 9% of articles are accessed directly, and the portion of traffic arriving from the rest of the Web is negligible. This documents how Wikipedia has become a well known and relevant resource and is prominently ranked by search engines for a diverse set of queries. About 30% of the traffic originating in Wikipedia is outbound, attesting to Wikipedia's important role as a reference to further information resources. The data, not surprisingly, show that the traffic to external resources is evenly spread among a large number of hosts, although the specific targets appear to be strongly biased by the user population of Indiana University affiliates. The 70% of internally directed traffic is evidence for the self-referential nature of Wikipedia.

Information about the origin and destination of Wikipedia traffic offers an opportunity to infer the usage mode for specific pages, as shown in Figure 5.3. This figure displays a heat-map of all Wikipedia pages. Their position along the two axes is determined by the amount of externally originated traffic they receive, and the amount of externally bound traffic they originate. I refer to this kind of plot as a *usage map*. Pages being represented in the upper left quadrant indicate a directory-like usage, with traffic mostly coming from inside and immediately leaving to outside resources. I interpret the upper-right quadrant as 'search' usage (pages visited mainly for the purpose of finding external resources), the lower right quadrant as 'encyclopedia' usage (pages visited from outside and leading to other internal resources), and the lower left quadrant as 'browsing' usage (from one internal page to another). With this interpretation in mind, Figure 5.3 suggests

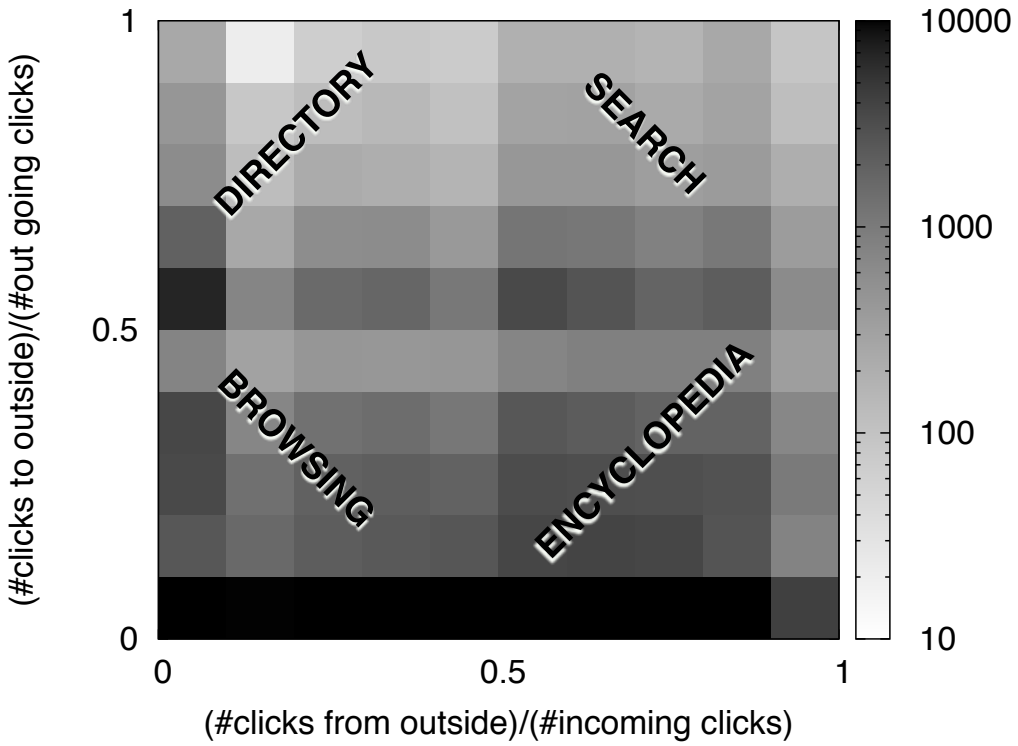


FIGURE 5.3. Usage map of Wikipedia pages. The X and Y axes represent the fraction of a page's traffic that comes from outside, or departs to outside, respectively. The shading of each bin represents, on a log scale, the number of pages with the corresponding usage. See text for an interpretation.

that while Wikipedia is used in all of these modes, the predominant usage modes are 'browsing' and 'encyclopedia,' as one might expect.

A related question to be asked is how "sticky" are articles in Wikipedia; that is, given that a user visits an article, how likely are they to click through to another article, versus clicking an external link? I cannot answer this question directly, as the traffic data does not track individual users, but rather aggregate behavior. Thus, I use the weighted graph induced by the traffic data to compare the weights of edges going from a Wikipedia node to another Wikipedia node, and compare these weights to the weights of edges going from the same source node to destination nodes outside Wikipedia. I use these weights to compute sample conditional probabilities for each action, yielding the probabilities shown in Figure 5.4. Generally, I may further conclude from this

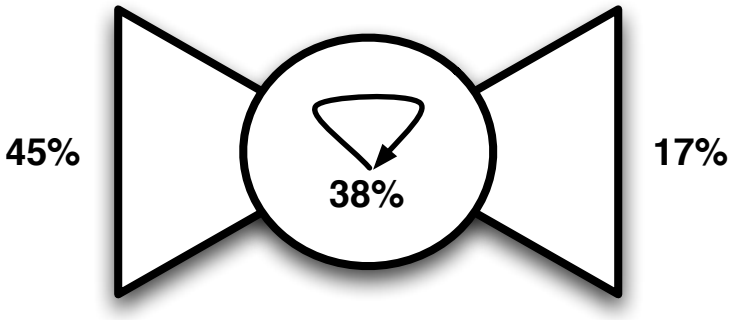


FIGURE 5.4. Probabilities of user movement patterns involving Wikipedia articles. Of the clicks in the data set, 45% represent users arriving from the outside Web to Wikipedia. 38% represent users moving from one Wikipedia page to another, and 17% represent users navigating from a Wikipedia page back to the Web.

data that Wikipedia is a sink, as the volume of traffic flowing out of it is about 30% less than that flowing in.

A finer level of resolution in this line of questioning can be attained by looking at the stickiness of individual *categories* of pages. Many Wikipedia pages are assigned these categories — more like tags than like any kind of hierarchy — by their human editors. By aggregating the previously-described conditional probabilities at the category level we can obtain lists of the most and least sticky categories, Here, we consider the event that a user stays within a particular category to be a Bernoilli trial, and compute the confidence interval for the probability of the success of this trial. Shown in Table 5.3 are the 95% confidence intervals for these success rates. I note that 'sticky' categories are categories for which a user browsing Wikipedia would tend to treat it like a hyperlinked encyclopedia, by following links to other pages within Wikipedia. When a category's stickiness is very low, this indicates that users tend to treat pages in this category like directories — using them to find links to pages in the Web at large (cf. Figure 5.3).

TABLE 5.3. Least (top) and most (bottom) 'sticky' categories.

title	**clicks**	**stickiness**
data structures	5133	[0.0, 0.02]
programming constructs	5127	[0.0, 0.01]
persistence	5106	[0.0, 0.01]
articles with example c code	2991	[0.0, 0.03]
stdio.h	2257	[0.0, 0.02]
male reproduction	11794	[0.02, 0.04]
italian-language operas	2370	[0.03, 0.06]
french-language operas	1364	[0.01, 0.05]
free software culture and documents	1361	[0.01, 0.05]
c headers	1318	[0.0, 0.04]
...	...	...
place name disambiguation pages	3149	[0.97, 1]
2000s music groups	5584	[0.93, 0.95]
grammy award winners	5285	[0.93, 0.96]
1990s music groups	4253	[0.94, 0.96]
greek mythology	3239	[0.94, 0.96]
self-organization	2582	[0.95, 0.98]
former british colonies	2241	[0.92, 0.95]
1980s music groups	2101	[0.95, 0.99]
surnames	1941	[0.96, 1]
former spanish colonies	1939	[0.92, 0.96]

5.1.2. Comparison with Other Networks

It is informative to compare Wikipedia usage patterns with those of other information networks as done in the previous subsection. To do this, I again leverage the traffic data (4.4) by selecting the records whose referring or target host is one of the following:

(1) The social networking site Facebook (`facebook.com`), used by many Indiana University students, staff, and faculty.
(2) The Indiana University Knowledge Base (`kb.iu.edu`), a hyperlinked technical reference site for the IU community that also provides general information of interest to outside users.
(3) The Google search engine (`google.com/search`).

For each of these sites, I constructed the weighted graph induced by traffic to and from their pages, during the same date range used for the Wikipedia traffic. I ignored requests for subordinate elements like images and advertisements, identified based on file extensions and known ad networks. For the Google and Facebook networks, I removed query strings from all URLs to avoid proliferation of seemingly unique URLs. Figure 5.5 shows the distributions of node degree and traffic (to or from a node). The largest network is Facebook, followed by Google and the Knowledge Base. In all cases I find very broad distributions of degree and traffic, in agreement with studies that have reported analogous properties for the Web at large [**MMF$^+$08**].

Figure 5.6 shows the usage maps for the three networks mentioned above and, for comparison, Wikipedia. Compared to the latter, one sees less encyclopedia and more directory usage in Facebook (from users posting external links), as well as a strong browsing component. I also observe that there is more traffic from Facebook to the rest of the Web than in the other direction. The Knowledge Base is used mostly as a proper encyclopedia, with the majority of outgoing traffic being directed to other internal pages. Finally the usage map for Google is the only one with a clear peak in the search quadrant. These observations suggest that usage maps can be a useful visualization tool for how a Web site channels human attention.

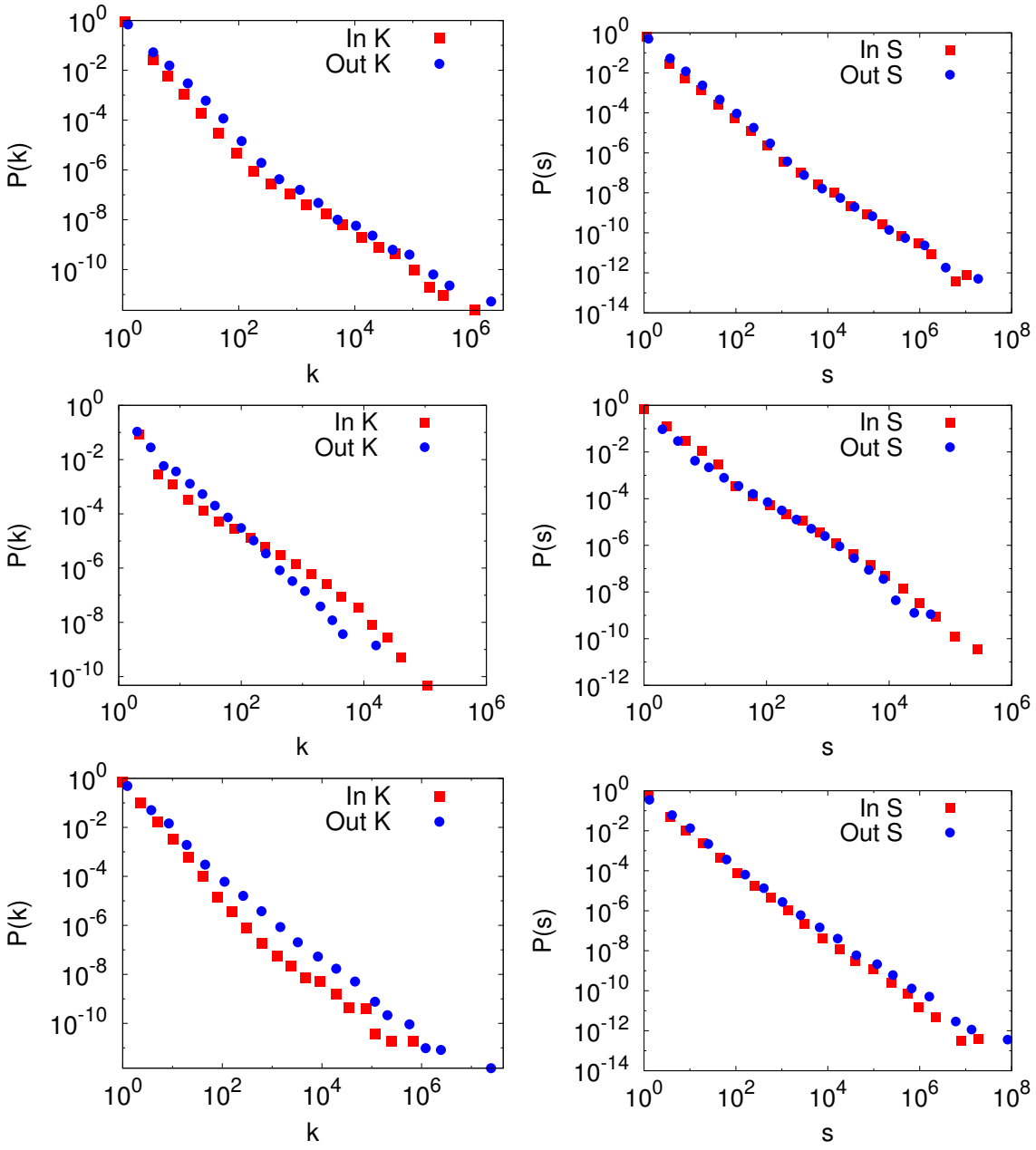


FIGURE 5.5. Distributions of degree (left) and traffic (right) for the Facebook (top), Knowledge Base (middle), and Google query (bottom) networks.

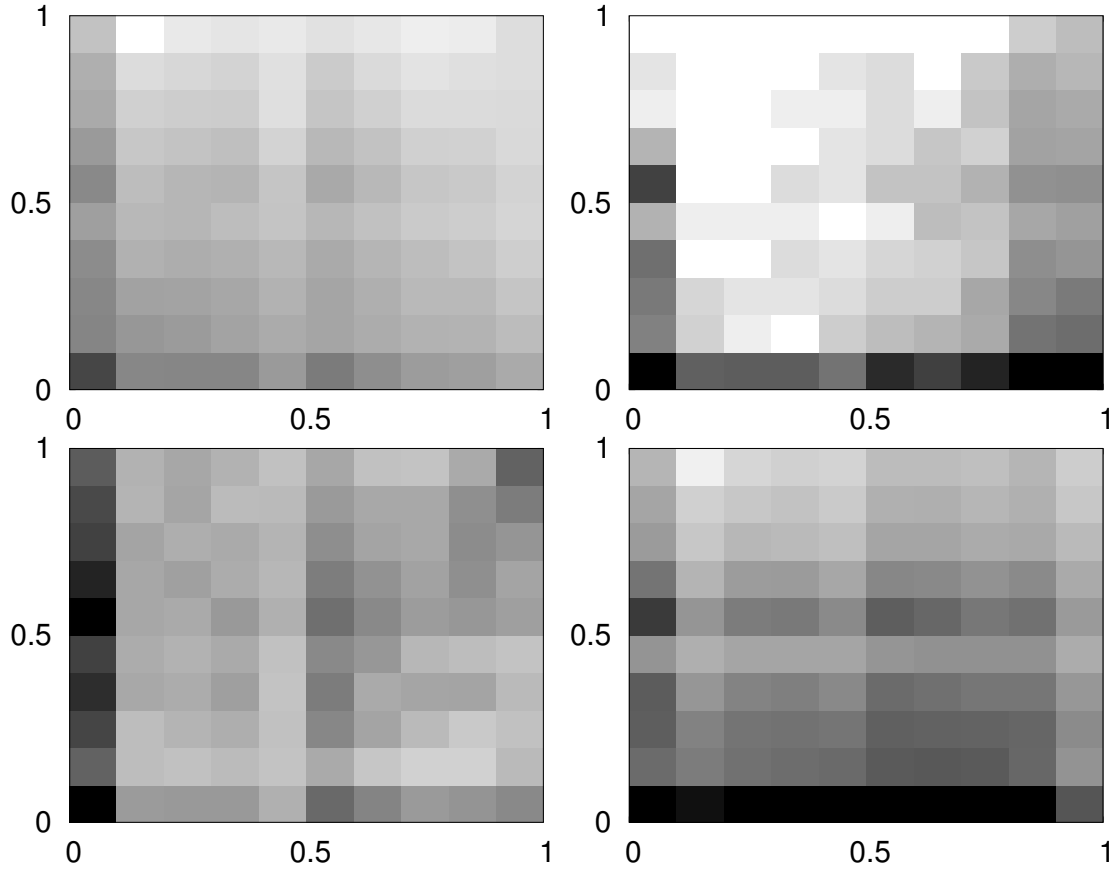


FIGURE 5.6. Usage maps for Facebook (top-left), Knowledge Base (top-right), Google query (lower-left), and Wikipedia (lower-right), visualizing the different modes in which pages in each of these networks are used.

5.1.3. What Drives Burstiness?

Beyond the above analysis of Wikipedia traffic, the *hits* data (4.3) offer a unique chance to take a step back and explore what may trigger users' interests in the first place. In this section I focus in particular on large deviations from *normal* traffic for specific topics. The peculiar distributions of size and frequency for these traffic bursts are characterized and modeled in Chapter 6.

It is natural to attribute these bursts of activity to real world events, possibly reflected in the news, that trigger the sudden interest of a considerable number of people in a short time span. The analysis described here aims to test this hypothesis by measuring the correlation between the appearance of news on a specific topic and sudden increases in traffic to the Wikipedia page on

that topic. I selected first the 200 most bursty articles, where the 'burstiness' of a page is defined as the ratio of its present to previous-day traffic averaged over the time span of the data. I disregarded pages whose present-day traffic was smaller then a threshold (set to 50 hits) to avoid noise fluctuations in traffic. I then constructed queries for each of these pages by removing stopwords and parenthesized words from the page titles; thus "Joe Wilson (U.S. politician)" became "`joe wilson`," and "Army for the Liberation of Rwanda" became "`army liberation rwanda`." These queries were then submitted to Google Trends, and the resulting search volume saved. It should be noted that this normalization process did not, in all cases, produce meaningful query strings; I refrained from correcting these cases by hand to avoid introducing bias. This process resulted in the construction of 200 Google Trends weekly time series. I then computed the Pearson correlation r between each Wikipedia topic's traffic and the Google search volume of its associated query.

The results are shown in Figure 5.7, combined with those of an analogous experiment focusing on the 200 most visited (rather than most bursty) pages. In this figure, the probability density function of the correlation r for the bursty pages clearly shows two peaks; one around zero, representing bursty pages with weak or no correlation with search volume data, and another closer to one representing pages with strong correlation. I hypothesize that the first of these peaks consists of pages that accrete traffic due to internal Wikipedia dynamics; these are explored in the next section. The second peak is clearly due to pages that suddenly receive large amounts of traffic due to news and world events. The distribution of r for the most visited pages, however, is more uniform between zero and one. This indicates that popular topics are more weakly correlated with search volume, with a smaller peak around one indicating that people may search for the same sorts of popular things on Google as on Wikipedia.

5.2. Microscopic Properties

We have seen that external events are directly responsible for triggering a large portion of Wikipedia traffic bursts. Let us now explore the dynamics by which users move within Wikipedia, and how they relate to the structure and content of the information network.

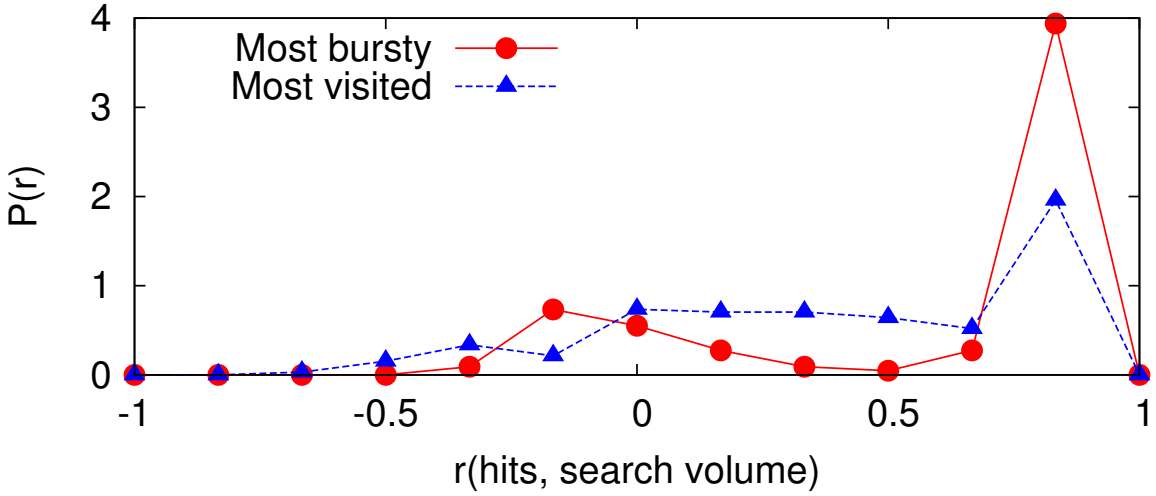


FIGURE 5.7. Probability distribution of the Pearson correlation between the traffic counts of the top 200 most bursty or most visited pages, and their associated search volume from Google Trends data.

5.2.1. How Pages Compare

Preliminarily, I examined the Pearson correlation between the time series of hits for pairs of pages satisfying various conditions. Each experiment was duplicated for weekly and daily time resolutions. I found the resulting distributions to be approximately normal; in all discussion below, the normal fits mentioned have $R^2 \geq 0.8$. For example, Figure 5.8 shows the distribution of the correlation r between pairs of pages, together with their best normal fits (computed by maximum likelihood). Note that the correlation is lowest between random pairs of pages, becoming higher when only linked pairs of pages are considered. Given these normal distributions, let us compare the traffic correlations by focusing on their means. Table 5.4 reports the estimated means for three hits correlations: between a page and a neighbor (i.e., a page connected by an incoming or outgoing link), between a page and its most correlated neighbor, and between a page and another page randomly selected from the whole Wikipedia. I report the results for the entire data set, as well as for a subset including only the 20% of pages with the most hits. This restricted data set accounts for over 90% of Wikipedia's total hits. All differences are significant at the 99% confidence level, with confidence intervals smaller than the least significant digits shown. I observe that pages are more correlated with their neighbors, and this effect is accentuated when we focus on the top 20% of

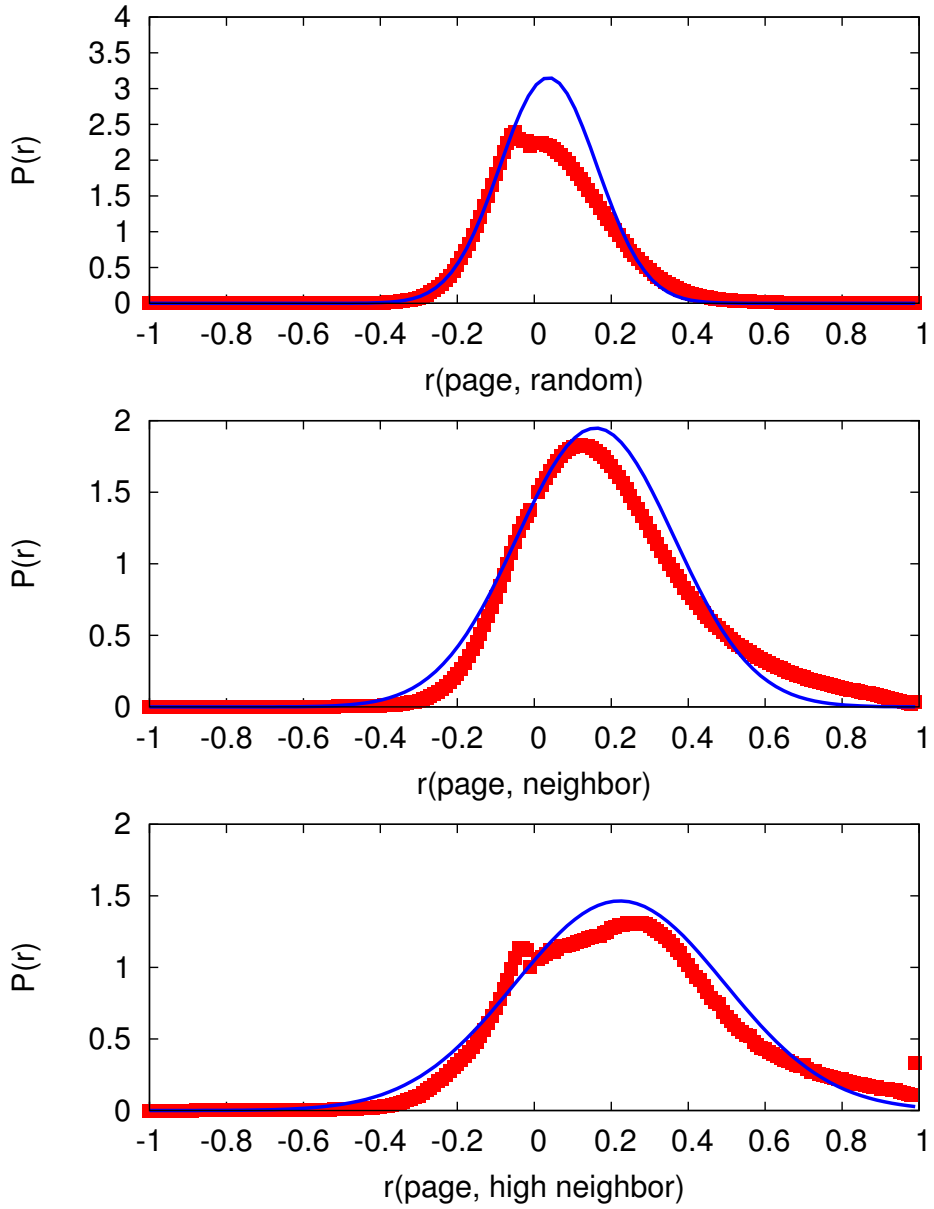


FIGURE 5.8. Distribution of the Pearson correlation r between pairs of pages at a daily time scale over two months, overlaid with their best normal fit ($R^2 \geq 0.8$). The discontinuities at 0 represent high peaks. The pair correlations shown are between random pairs of pages (top), linked pairs of pages (middle), and between pages and their highest-correlated neighbors (bottom).

TABLE 5.4. Mean Pearson correlations between hits time series.

		All pages	**Top 20%**
Daily	r(page, high neighbor)	0.22	0.52
	r(page, neighbor)	0.16	0.29
	r(page, random page)	0.04	0.05
Weekly	r(page, high neighbor)	0.46	0.68
	r(page, neighbor)	0.27	0.35
	r(page, random page)	0.25	0.31

pages (thus eliminating pages that are visited infrequently). Further, note that the increase in correlation between pages and neighbors versus random pairs of pages all but disappears for weekly time resolution. This indicates that the weekly time scale is so large as to smooth over interesting features in the data; therefore, I omit it in further analysis in favor of the daily time scale.

5.2.2. Why Neighbors are Correlated

We now know that neighbors are correlated in the hits that they receive. The observation that two neighboring pages (say a and b) experience similar levels of traffic is consistent with the following two scenarios:

(1) Pages a and b are topically similar, and external factors generate interest in their common topic; as a result, both pages experience similar levels of traffic.
(2) One of the pages, say page a, sends a large portion of its traffic along its link to page b, causing their levels of traffic to be similar.

To tease apart these effects, I performed several more experiments. The first was to look at the distribution of content similarity among linked versus random pairs of pages; the results of this experiment are shown in Figure 5.9. We see that linked pages are far more likely to be similar than randomly chosen ones. When I consider for each page its neighbor with highest hits correlation, I find that the similarity tends to be higher still. Further, I produced a scatter plot representing the relationship between hits correlation and content similarity among linked pages; the result is shown in Figure 5.10. I find that in general, there is a very weak (but non-zero) correlation between the traffic and content similarity of linked pages.

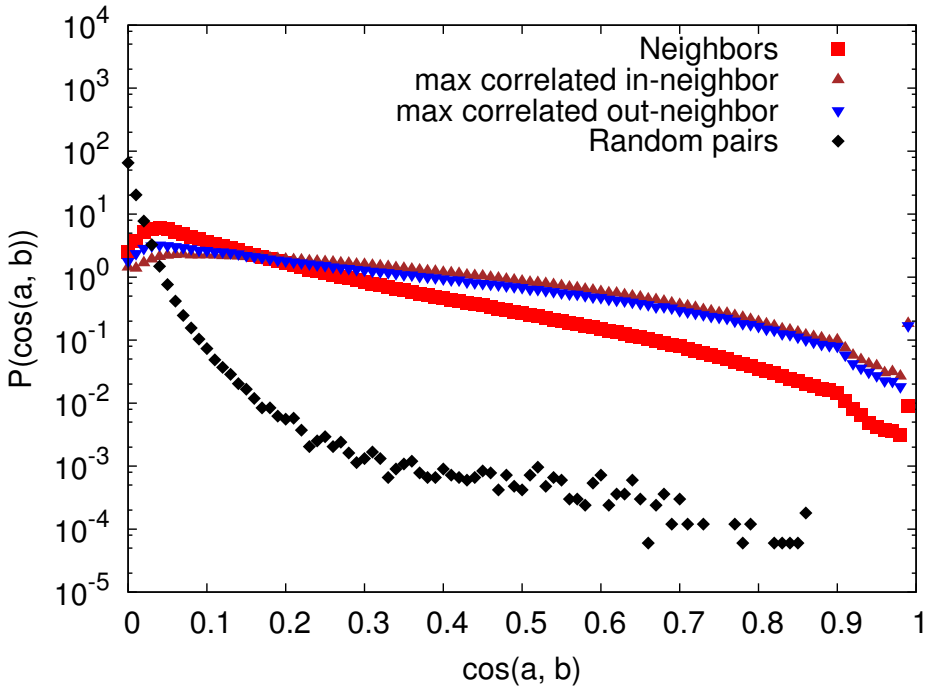


FIGURE 5.9. Distribution of the cosine similarity between pairs of pages selected according to various criteria. The two curves above the neighbor distribution represent the distribution of cosine similarity between a page and specific neighbors; namely, those (in/out) neighbors that have the maximum Pearson correlation (r).

To determine the influence of traffic flowing across links between pages, I need the additional information provided by the traffic data set (4.4). I want to see how much of the correlation between the traffic received by two linked pages is due to direct traffic from one to the other. Let $s(a)$ and $s(b)$ be the time series of traffic to topics a and b. Let $s(a \to b)$ be the direct traffic from a to b. Figure 5.11 shows a scatter plot of the correlation between $s(a)$ and $s(b)$, versus the correlation between $s(a)$ and $s(b) - s(a \to b)$. Points near the diagonal therefore represent pairs of topics whose traffic correlation is not explained by direct traffic between them (scenario 1). Points along the x axis represent pairs of topics whose traffic is no longer correlated when direct traffic is removed (scenario 2). Based on the traffic data, this latter scenario is predominant. In other words, traffic from a to b causes in many cases the correlation in traffic between a and b.

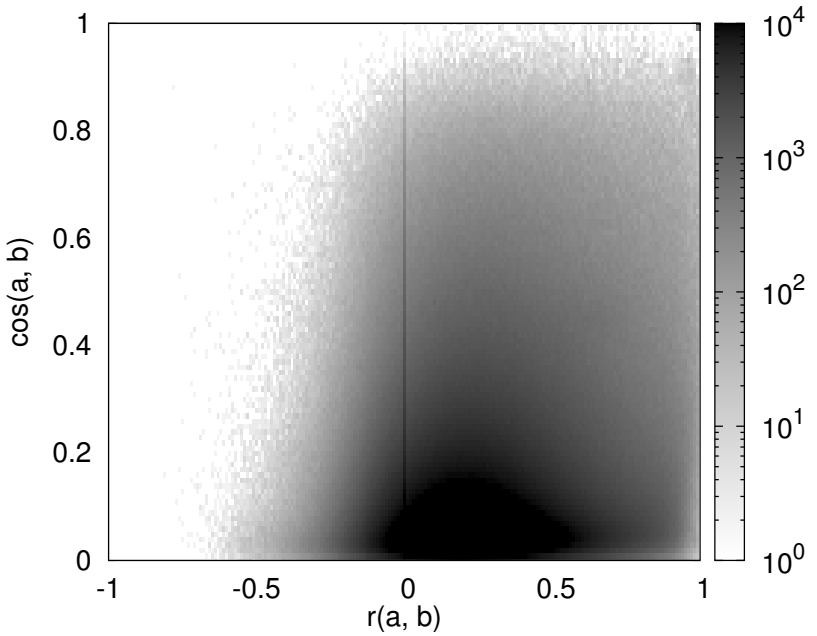


FIGURE 5.10. Heat map visualizing a scatter plot of the the Pearson correlation $r(a, b)$ between the hits time series of linked topics a, b at a daily time resolution, versus the cosine similarity $\cos(a, b)$ between their TF-IDF vectors.

5.3. Application: Wikipedia Category Prediction

As a potential application of the type of analysis presented here, let us explore some simple techniques for predicting categories of Wikipedia pages — tags assigned by editors. The task is as follows: for the subset of pages that *(a)* are in the top 20% of pages by hits, *(b)* have at least one human-assigned category, and *(c)* have at least one out-neighbor, use the category assignments of a page's out-neighbors to predict its categories.

Given the category assignment matrix C, where $c_{\chi,p} = 1$ iff category χ has been assigned to page p, I apply a modified nearest neighbors algorithm:

For each page p in our set:

(1) Rank p's neighbors by some similarity score (see below). A fraction f of the neighbors will be allowed to vote on p's categories.

(2) Let C_p be the union of the sets of categories assigned to each of p's neighbors. Compute a vote weight w_χ for each $\chi \in C_p$ defined as the number of neighbors of p that are assigned category χ.

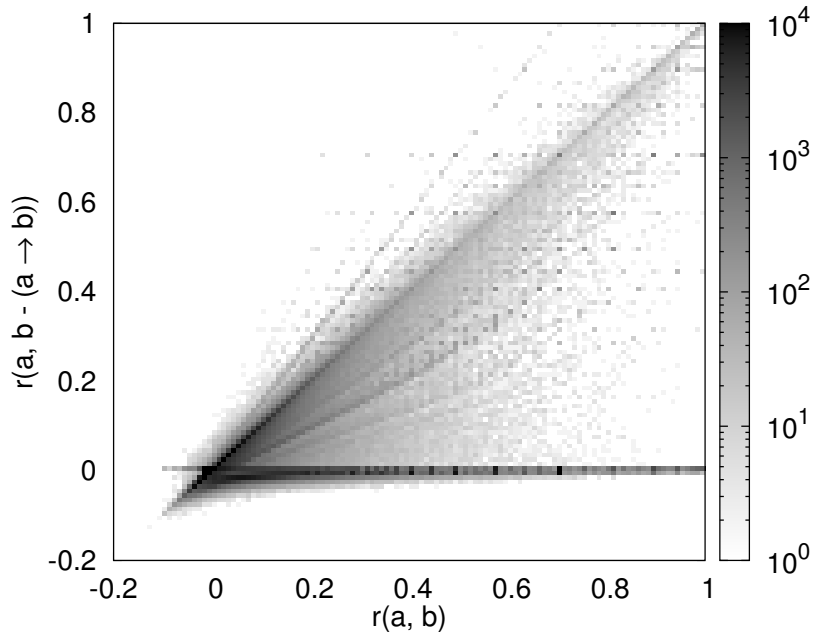


FIGURE 5.11. Heat map visualizing a scatter plot between the Pearson correlation between two page's daily traffic (x axis), and that same traffic when the traffic traveling directly from the first to the second, via a link between them, has been removed (y axis).

(3) Rank the categories according to the weights w_χ, so that χ_r is the rth category. Evaluate by Mean Average Precision (MAP):

$$\frac{\sum_{r=1}^{|C_p|} P(r)\; c_{\chi_r,p}}{\sum_\chi c_{\chi,p}} \tag{21}$$

where $P(r)$ is the precision at rank r, i.e., the fraction of the top r predicted categories that are correct.

We experiment with three ranking methods for the neighbors q of page p in step (1) of the algorithm: *(i)* the cosine similarity $\cos(p,q)$, *(ii)* the hits correlation $r(p,q)$, and *(iii)* the actual traffic $s(p \to q)$. Further, to bound the results, we add *(iv)* a random ranking, and *(v)* a greedy ranking by the size of the overlap between the category sets of p and q. Note that the algorithm based on ranking *(v)* assumes knowledge of the categories of p and therefore is not a proper predictor. The results are shown in Figure 5.12. The method that ranks neighbors by their cosine similarity outperforms all others, achieving a peak MAP for $f \approx 0.2$ before tapering off as less relevant

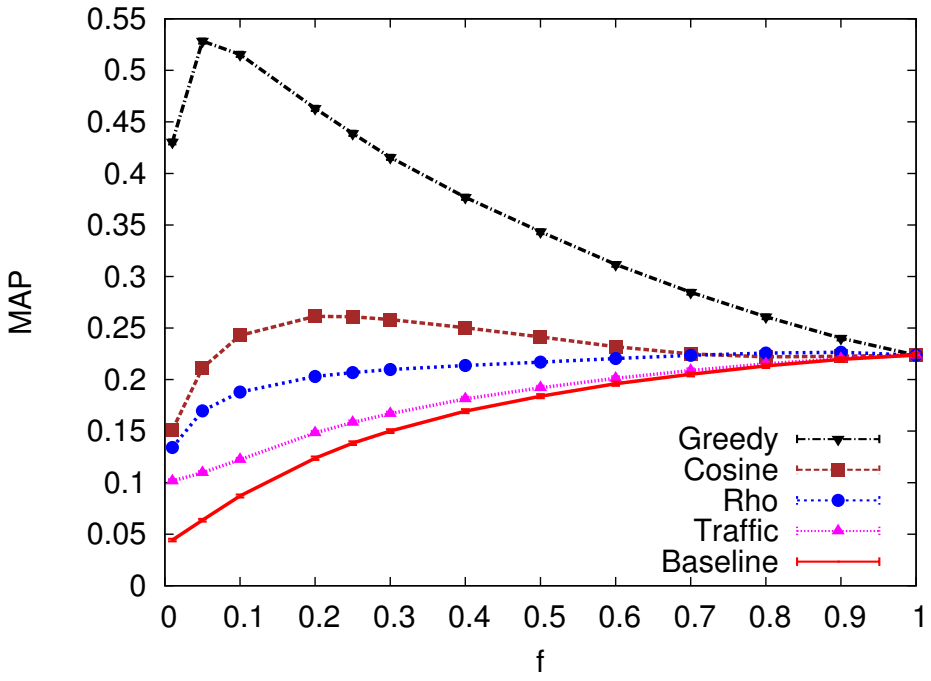


FIGURE 5.12. Mean Average Precision for recovering categories of a Wikipedia page, as a function of the fraction of neighbors allowed to vote for the predicted categories. Error bars for a 95% C.I. are shown, but are so small as to be obscured by the points.

neighbors are added. The methods based on correlation r and traffic outperform the baseline, but do not perform as well as content similarity; however, the comparison with the greedy algorithm suggests that all algorithms could be improved. I leave as a topic for future research the question of how to combine these ranking methods to improve their performance.

5.4. Summary

This chapter presents the results of a major longitudinal study of Web traffic data, across several sites and gathered from several sources. The data are combined to provide a synthesis of Wikipedia usage by real Internet users. My approach allows for the development of a high-level understanding of the position Wikipedia has with respect to the Web at large; where users come from, and where they go. Further, I introduce a simple graphical visualization (the *usage map*) capable of giving a high-level picture of how pages in a network tend to be used, providing us with

a key to interpret the way in which the network itself is navigated. This visualization makes precise some intuitions about how, for instance, the usage of pages on Wikipedia differs from that of pages on Facebook. Further, I find that pages that experience sudden bursts of traffic in Wikipedia often correspond to topics that have attracted sudden bursts of attention in the Web at large, as measured by Google search volume. Results from a number of experiments addressing how users move between pages in Wikipedia are presented. I conclude that users tend to move between pages in some correlation with their content similarity, and that high traffic correlation among neighbor pages is often caused by direct traffic between them. Finally, I tried to exploit similarity in content and traffic among topics to predict Wikipedia page categories. Methods based on traffic fail to outperform those based on content, but there is plenty of room for improvement even in content-based methods; future work could explore ways of combining these methods.

CHAPTER 6

MODELING BURSTS IN ATTENTION

6.1. Introduction

Following the general exploration of popularity behavior outlined in the last chapter, let us now turn to the problem of studying the growth of popularity *longitudinally*. While many growth models have been designed to model the state of a graph at a fixed point, there are relatively few that match real-world networks in the *dynamics* of their growth, as well as the end result. This chapter outlines some efforts in understanding this problem. I begin by the introduction of an analytical tool, the *logarithmic derivative*, which enables studying the growth of a system irrespective of its size — thus patterns of growth can be compared between systems of different sizes. I then apply this to several large-scale longitudinal data sets, finding some remarkable similarities in the growth of systems of different sizes and with different human audiences. I also outline the intriguing similarity between bursts of attention in these systems, and similar 'bursty' behavior in the real world, like avalanches and earthquakes.

6.2. Methodology

I analyze three large scale data sets about two information networks for which it is possible to gather longitudinal information: the entire Wikipedia and the Chilean Web. The nature and preprocessing of these datasets is described in Chapter 4, § 4.1 and § 4.2. I also use the hits data for the Wikipedia described in Chapter 4 § 4.3. Table 6.1 is a reminder of the sizes of these datasets.

An initial issue facing a study of the sort presented in this chapter is the identification of a suitable popularity measure. In recent years, the mapping of large, complex information networks [**AJB99, BKM$^+$00, SMB$^+$07**] has led to identifying the number of links pointing to a node (its *indegree*) as a proxy of popularity in many domains [**SP06**]. The evidence that many social, technological, and information networks are characterized by stable heavy-tailed distribution of indegree pointed to a strong heterogeneity in the popularity and triggered the formulation of models aimed at explaining the emergence of such broad distributions using rich-get-richer mechanisms [**Sim55**] based exclusively on topology [**dSP76, BA99, KKR$^+$99**] or combined with content information [**Men04**]. While these models have the merit of introducing irreversible growth as an important element of network generation, the dynamics characterizing these rapidly changing systems have been seldom studied because to date it has been infeasible to observe the actual growth of an online network. The datasets I utilize, however, contain longitudinal information that makes it possible to observe their growth. Further I have access to traffic data, which I consider a more direct proxy to popularity as it represents human attention more immediately.

In both the data sets (English Wikipedia and Chilean Web) I track the time evolution of the indegree k of documents, i.e., hyperlinks from other Wikipedia articles and other Chilean Web pages, respectively. In Wikipedia the high temporal resolution allows me to analyze this measure as a function of real time or age since the creation of a page, and using different timescales — e.g. months, weeks, or days — over the entire edit history. For the Chilean Web I can track the indegree with the time resolution of a year. In Wikipedia I also track the number of times s that an article is actually visited; traffic is a more direct measure of the interest generated by each topic.

With each of the three data sources — Chilean Web, Wikipedia articles, and Wikipedia traffic counts — I produced a matrix in which the rows correspond to nodes and columns to dates, with each entry in the matrix referring to the value of the popularity measure for that node and date.

TABLE 6.1. Summary of datasets. Each is represented as a matrix in which the rows represent individual nodes, and columns represent dates.

	Vertices	First	Last	Temporal Resolution
Wiki k	3 293 102	Jan 2001	Mar 2007	1 sec.
Wiki s	3 490 740	Feb 2008	Current	1 hour
Chile k	3 252 779	2001	2006	1 year

For the Chilean Web and Wikipedia articles this popularity measure was indegree (k); for Wikipedia traffic it was incoming traffic, s. Details on all preprocessing can be found in Chapter 4.

6.2.1. Measures

To quantitatively study the dynamics of any time dependent popularity measure x_t, it is convenient to consider its *logarithmic derivative*

$$\Delta x/x_t = \frac{x_t - x_{t-1}}{x_{t-1}}, \tag{22}$$

where t refers to units of time. This allows us to compare the dynamics of pages with different popularity while discounting the overall growth of the underlying system, which is not uniform across data sets. Figure 6.1 illustrates the logarithmic derivative of the indegree of two example pages in the English Wikipedia. Despite a roughly exponential growth in the popularity of both topics, the logarithmic derivative provides a signature by which the two profiles can be compared on the same scale. Almost all pages experience a burst in $\Delta x/x$ near the beginning of their life,[1] and many receive little attention thereafter. While some pages maintain a nearly constant positive logarithmic derivative indicating an exponential growth, a number of pages continue to experience intermittent bursts in $\Delta x/x$ later in their life.

6.3. Results

6.3.1. Burst size distribution

As a first step, we confirmed scale-free distributions of popularity in our Wikipedia data, finding each (both indegree and traffic) to be well modeled by a power-law distribution. This is in agreement with other studies and with results for the Web at large. We know from Baeza-Yates and

[1] We ignore the initial step of a page's life, where $x = 0$ and $\Delta x/x$ would be undefined.

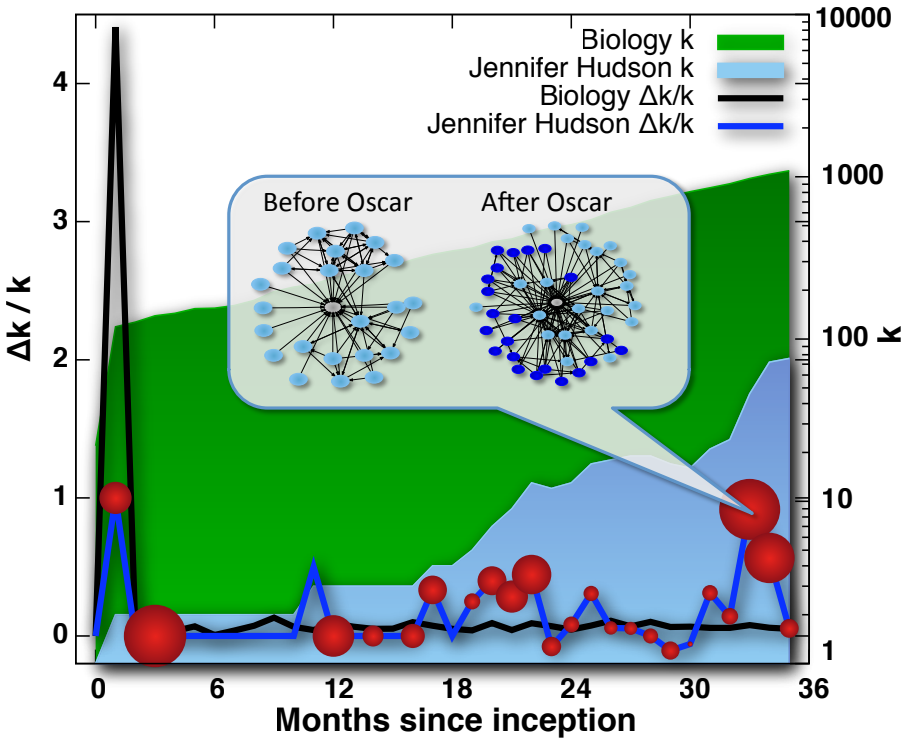
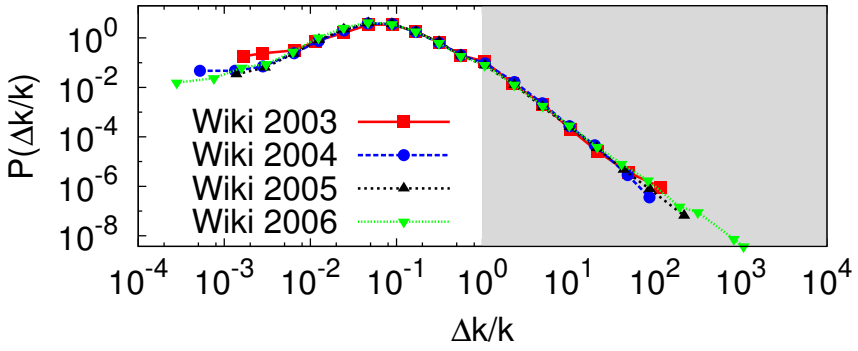


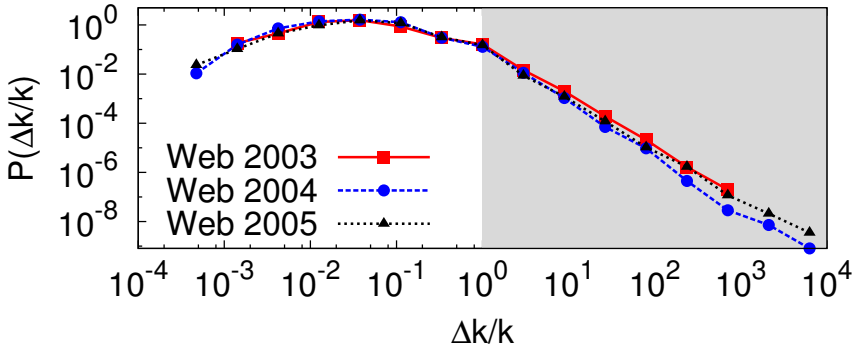
FIGURE 6.1. Time series of indegree k and its logarithmic derivative $\Delta k/k$ for two Wikipedia topic pages. Topics typically experience a burst in their early life. The 'Biology' page then maintains a small rate of growth. The article about Jennifer Hudson, however, experiences more fluctuations later in its life. Jennifer Hudson is an artist who became popular through a television show leading to her first burst. Another burst occurred when she won an Academy Award; degree popularity doubled as many other pages linked to the article (inset). Another popularity measure is also shown for the 'Jennifer Hudson' page; the size of each circle is proportional to the logarithmic derivative of the number of times the article is revised. The article receives more edits when it attracts more links.

Poblete **[BYP06]** that this is also true in the Chilean Web data we study. We next turn our attention to the distribution of the log derivative of popularity $\Delta x/x$.

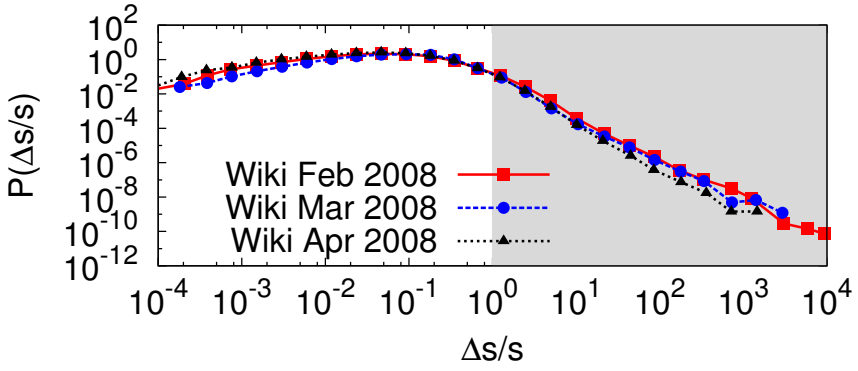
The distribution of the magnitude of $\Delta x/x$ for the two popularity measures at representative time resolutions is illustrated in Figure 6.2. All curves provide striking evidence for a wide variability of the burst magnitude that spans 8 or 9 orders of magnitude. In all cases and at all granularity it



(a) $\Delta k/k$ for Chilean Web indegree, with temporal resolution of one year.



(b) $\Delta k/k$ for Wikipedia indegree, with temporal resolution of one month, as measured in January over several years.



(c) $\Delta s/s$ for Wikipedia traffic, with temporal resolution of one week, as measured over a few months in 2008.

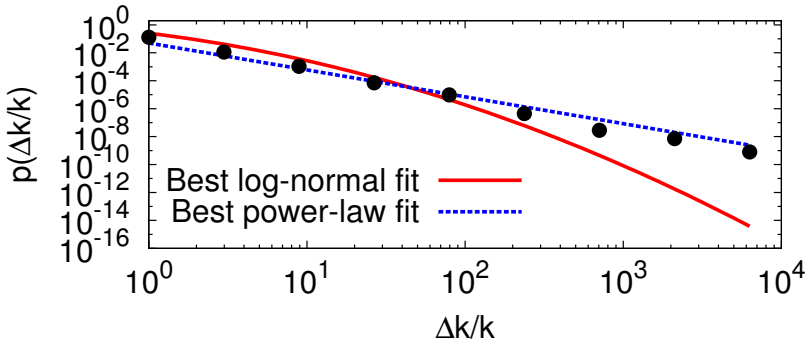
FIGURE 6.2. Distributions of logarithmic derivative of popularity. In each plot, the gray area highlights the power-law tail of the distribution. These behaviors are consistent across a wide range of temporal resolutions, as observed using time units from a week to a year.

TABLE 6.2. Maximum-likelihood power-law and log-normal fits to each of the three datasets, with their Kolmogorov-Smirnov statistics. In each case the power-law fit outperforms the log-normal fit. The fits are computed for the tails of the distributions only, that is, for $\Delta x/x \geq 1$. Figure 6.3 shows the data plotted with these fits.

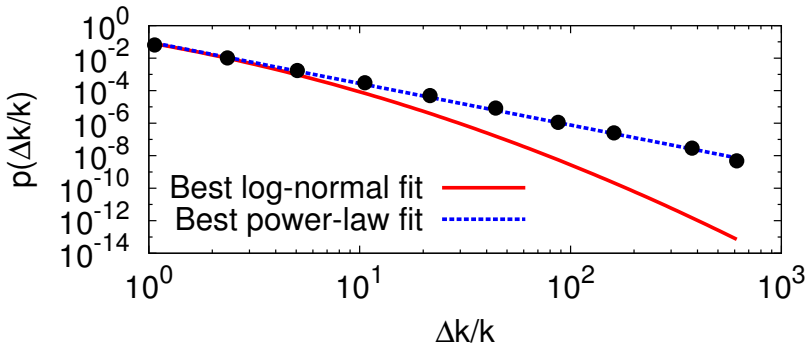
	Powerlaw fit		Lognormal fit		
	α	K-S	μ	σ	K-S
Chilean web $\Delta k/k$	1.924	0.008	-0.715	1.389	0.0997
Wikipedia $\Delta k/k$	2.559	0.012	-2.002	1.250	0.0358
Wikipedia $\Delta s/s$	2.134	0.038	-1.686	1.198	0.2015

is possible to observe a heavy-tail behavior for the statistical occurrence of large magnitude events. The observed long tails are stable, but are they well approximated by power-law dynamics, as opposed to more narrow distributions such as log-normal? To answer this question, I turn to the techniques described in detail by Clauset *et al.* **[CSN07]** (and outlined in Chapter 2 § 2.6.3) to compute maximum-likelihood power-law fits with $x_{min} = 1$ for each of the distributions of logarithmic derivatives. Note that by this I do not attempt to fit the 'hump' occurring for $x < 1$, but rather the long tail that is observed for $x \geq 1$. For comparison, I also compute the maximum-likelihood log-normal fits. Table 6.2 lists the parameters of all fits, together with the Kolmogorov-Smirnov statistic of each; the latter approaches 0 as the fit approaches the empirical data perfectly (see Chapter 2 § 2.5.4). Thus, in all cases the power-law fit outperforms the log-normal fit in modeling the long tail of the logarithmic derivative. The empirical data plotted together with these fits are shown in Figure 6.3.

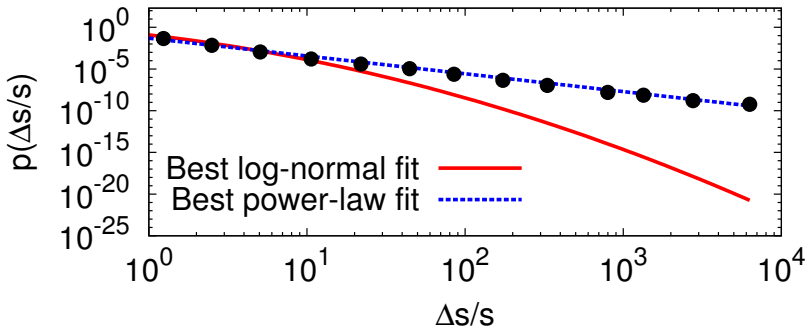
The performance of these power-law fits indicates that a statistically appreciable fraction of events corresponds to increases in popularity by factors of 10–10^3 or more. Such a disproportionate jump of interest occurs not only for young or lesser known pages, but for pages across a broad range of popularity. To illustrate this, I plot in Figure 6.4 the indegree of pages which undergo a burst in the following timestep, and compare to the indegree distribution of all pages. We see by this that even pages with large indegree can still experience dramatic changes. Yet additional evidence is that when pages below a certain age (e.g. 3 months) are ignored, the distributions in



(a) Chilean Web, time scale 1 year

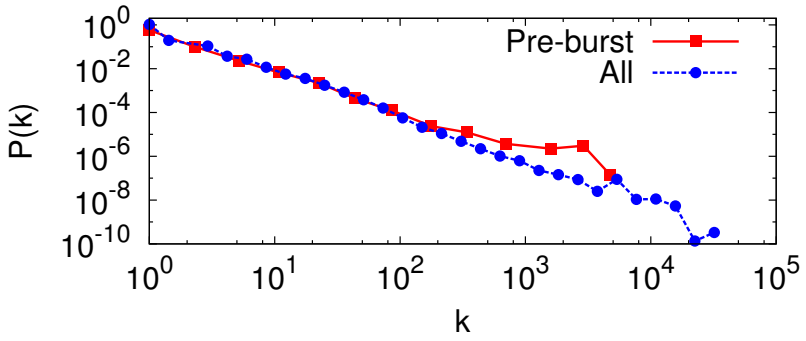


(b) Wikipedia degree, time scale 1 month

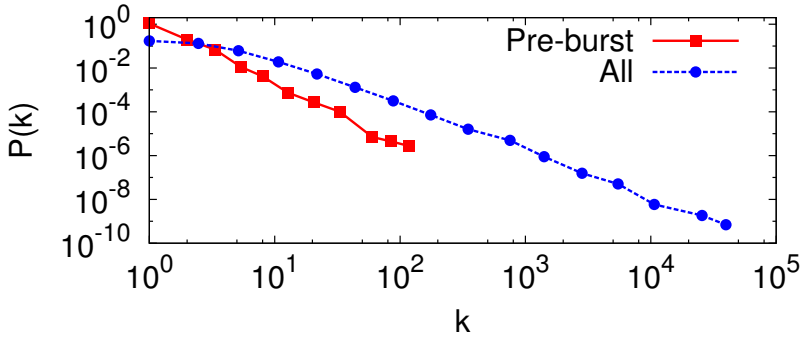


(c) Wikipedia traffic, time scale 1 week

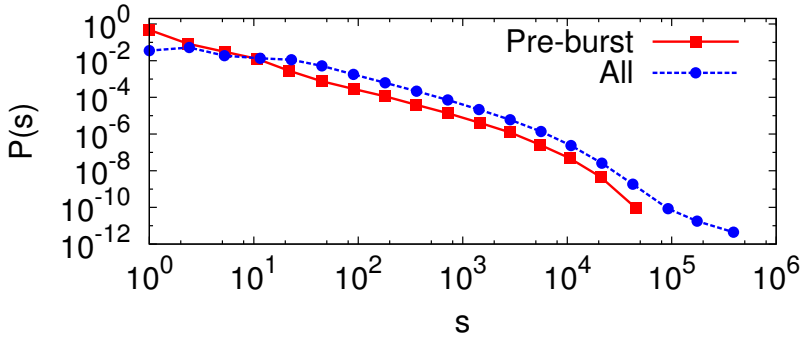
FIGURE 6.3. Maximum-likelihood power-law and log-normal fits for the long tail of the log-derivatives of the three data sets. The parameters and goodness for these fits are given in Table 6.2. In all cases the power-law fit outperforms the log-normal.



(a) Chilean Web k



(b) Wikipedia k



(c) Wikipedia s

FIGURE 6.4. Distributions of popularity x for pages with $\Delta x/x > 1$ in the subsequent timestep. The broad distribution of this value shows that bursts do not occur solely to young or unpopular pages.

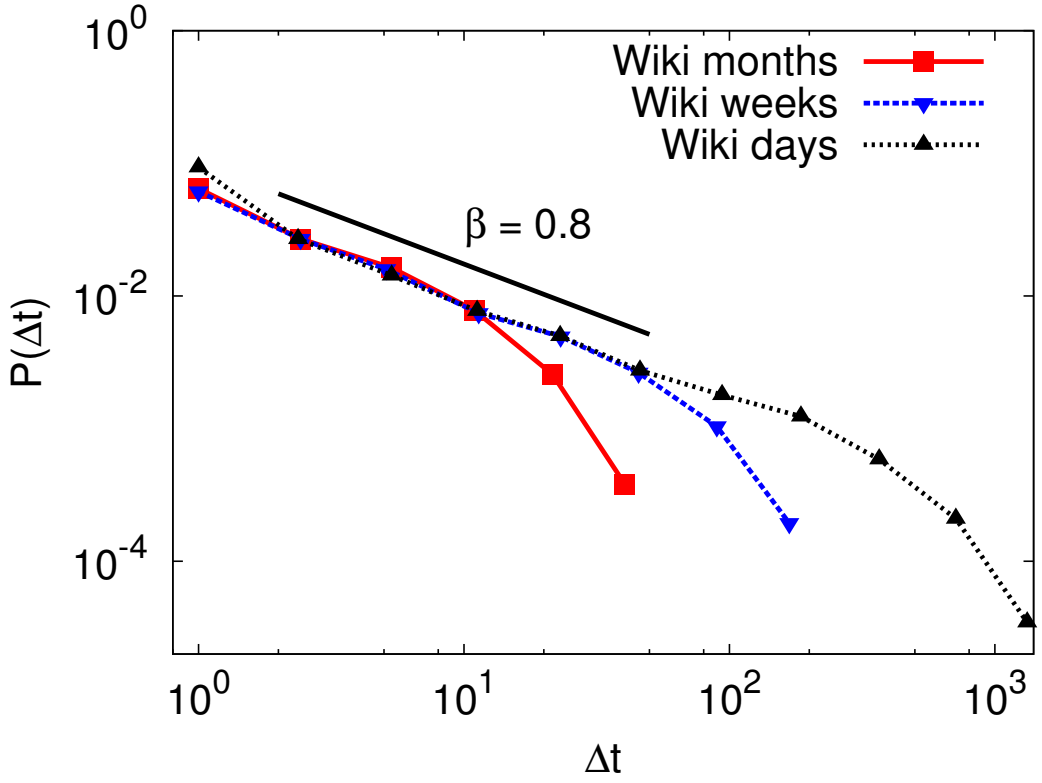
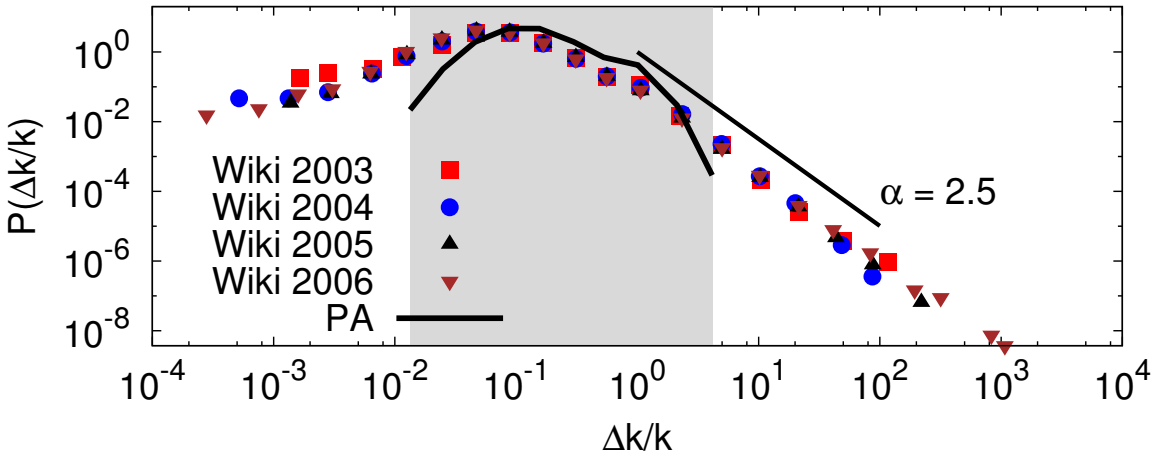


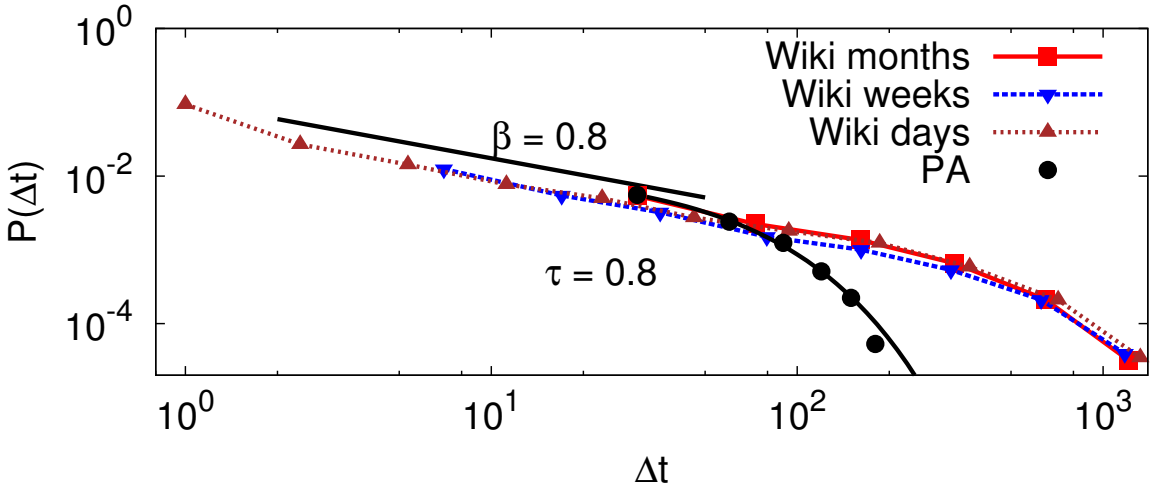
FIGURE 6.5. Distribution of the time interval Δt between consecutive indegree bursts of Wikipedia articles. The three curves correspond to different time resolutions of months, weeks, and days, aligned on the x-axis for ease of visualization. As we increase the resolution the tail of the distribution extends further, an indication that the cutoff is a finite size effect. As a guide to the eye I show a power law $p(\Delta t) \sim (\Delta t)^{-\beta}$ for $\beta \approx 0.8 \pm 0.1$.

Figure 6.2 are unchanged. In other words, these popularity spikes are statistically possible for all documents almost independently of their popularity.

These heavy-tailed burst magnitude distributions suggest a dynamics characterized by the lack of a typical scale for measuring the bursts. This is typical in a wide range of "critical" physical, economic, and social systems, such as avalanches, earthquakes, and stock market bubbles and crashes [**Bar05, Man97, SAB+96, GR44**].

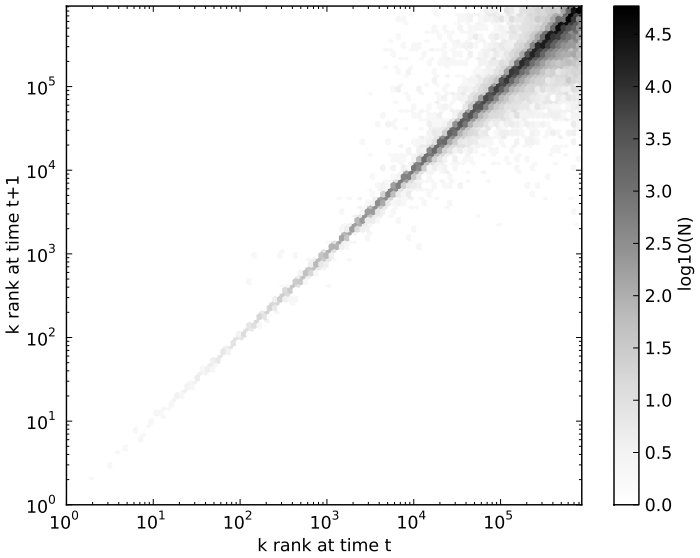


(a) Empirical distribution of $\Delta k/k$ from the English Wikipedia, together with analogous data as generated by a preferential attachment model. The long tail of burst sizes, highlighted by a power-law guide to the eye, is missed by the PA model, which generates data only in the gray area.

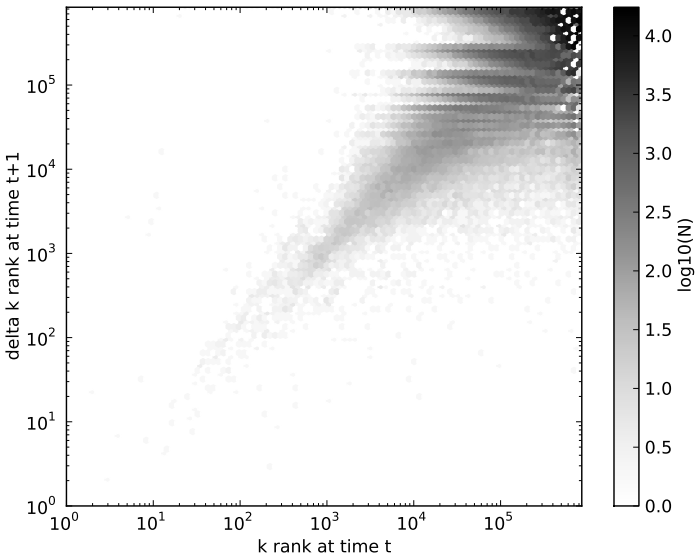


(b) Empirical distribution of time between bursts Δt (in normalized units) in the English Wikipedia, together with the distribution generated by a preferential attachment model. While the empirical data fits a power law (cf. Figure 6.5), the PA distribution fits an exponential $P(\Delta t) \sim e^{-\Delta t/\tau}$ with parameter $\tau = 0.8$.

FIGURE 6.6. Comparison of the empirical data with what would be expected from a preferential attachment process. The PA process fails to produce wide distributions of event size and temporal spacing.



(a) Rank of k at a representative time t vs. rank of k in the subsequent timestep.



(b) Rank of k at a representative time t vs. rank of Δk in the subsequent timestep.

FIGURE 6.7. Scatter plots visualizing changes in rank of k and Δk between timesteps. The presence of points away from the main diagonal indicates behavior other than that predicted by preferential attachment.

6.3.2. Distribution of time between bursts

Another way to characterize the dynamics of bursty systems is to study the distribution of times between successive events. In traditional systems where this behavior is modeled by queueing theory, we expect this distribution to be Poissonian. On the other hand, systems which lack a typical scale in the event size are generally associated with a lack of characteristic time scale and to long-range time correlation among consecutive events. To test for the presence of non-Poissonian dynamics I analyzed the time distribution between bursts, shown in Figure 6.5 for the English Wikipedia. I consider bursts such that $\Delta k/k > 1$ after January 1st, 2003. This necessarily includes pages which undergo smaller bursts (in absolute terms); e.g. pages whose popularity measure goes from 1 to 2. However, I observed that thresholding did not change the statistical properties of burst events — recall from Figure 6.4 that even pages with high popularity can experience large bursts. The intervals between bursts are broadly distributed in a power-law fashion with a finite size cut-off, as in Omori's law of earthquakes and other avalanche phenomena [**Omo94**]. The intriguing analogy between online popularity dynamics and critical avalanche phenomena calls for a stylized model able to explain the observed features in terms of shifts in collective attention. Critical avalanche processes with such scaling behavior are usually present in driven-dissipative systems where a quantity is introduced at a very slow rate and dissipated through a sudden non-linear mechanism [**VZ98**].

6.4. Modeling Popularity Trends

6.4.1. Preferential Attachment

Among the many growth models in the general family of preferential attachment, I chose the directed version [**DMS00**] of the linear preferential attachment model [**BA99**] as a baseline, and used it to generate a graph. This rich-get-richer mechanism does produce graphs with the same degree distribution as in our data sets; however, preferential attachment alone fails to reproduce the long tails observed in the distributions of both $\Delta k/k$ and inter-burst time (Figure 6.6).

Another way to explore the limitations of PA in explaining the observed dynamics is to visualize the relationship between the rank of a node's indegree k at a given time step, and its behavior in the time step that follows. In Figure 7(a) we show a scatter plot comparing a node's rank in k at

time t with its rank in k at time $t+1$. Figure 7(b) similarly shows the relationship between a node's rank in k at time t and its rank in Δk at time $t+1$. This visualization suggests that the empirical data has an underlying preferential attachment component, but with a strong chance of large changes, especially for nodes of lower degree.

These and other observations suggest the need for a model that also captures the sudden, dynamic changes in attention which I observe in the empirical data.

6.4.2. A Rank-Based Model

Seeking a very simple model able to capture the critical dynamics observed empirically, I note that the accumulation of attention is not obviously related to the exact degree of a document, information that is seldom available. Popularity is instead likely related to the relative ranking that is always established by users according to some criterion: age, degree, relevance to a user query (if the nodes are Web pages), or some arbitrarily distributed prestige function. I consider a generalization of the *ranking model* **[FFM06]** where items are sorted according to some popularity criterion and accumulate units of popularity such that the probability that an existing item i receives a unit is $p(i) \sim r_i^{-\delta}$, where r_i is the rank of i according to some arbitrary ranking function, and $\delta > 0$ is a free parameter. This simple model leads in the asymptotic limit to scale-free popularity distributions $p(x) \sim x^{-\gamma}$, where $\gamma = 1 + 1/\delta$. The behavior is very robust with respect to choices of the ranking criterion and of the exponent δ. Since popularity is distributed based on the ranks of the nodes, and not on their popularity values, the ranking model does not belong to the class of fitness-based models [**BB01, BS03**].

In this study of Web and Wikipedia pages I focus on the statistics of extreme events, represented by popularity "bursts." I define a burst as a variation of popularity Δx (within a given time window) larger than the original popularity value x of the page, i.e., an event with logarithmic derivative $\Delta x/x > 1$. The distribution of the time elapsed between consecutive bursts of the same node has a Poissonian decay for the ranking model, at variance with my empirical observations. Therefore, a modification of the model must be devised. Pending further study, I want the model to be agnostic to the actual cause of the bursts; in real data, they could be caused by external events (such as interest sparked by news stories) or due to other dynamics of the system (recall Figure 5.1).

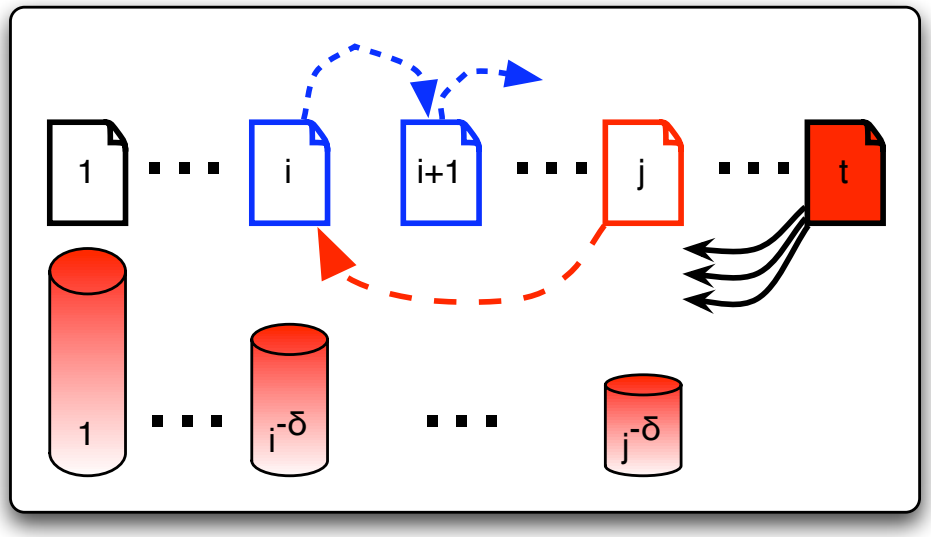


FIGURE 6.8. Illustration of the rank-shift model in an example where popularity is measured by indegree. New nodes are added at each timestep, illustrated by the node t; each node's probability of receiving a new link is proportional to their rank. In the diagram the node j is being re-ranked, pushing down the ranks of $i, i+1, ...$

For now, observe that the net effect of such a burst for the node in question is to change its popularity rank with respect to the other nodes in the system. Therefore, let us introduce *rank shifting* in the model: at each iteration, with a small probability ρ each node is assigned a new rank, chosen uniformly between 1 and its current rank, simulating a sudden increase in the attention paid to the node. (Figure 6.8). Thus this new *rank-shift* popularity model has two parameters: δ regulating the probability of accumulating popularity as a function of rank, and ρ defining the frequency of rank perturbations for each page. These sudden improvements of the rank lead to abrupt variations of popularity, as observed in the empirical data.

The model works as follows. Each node is assigned an arbitrary position in an initial ranking. Then two steps are performed iteratively. First, a new node t is added, and linked to existing nodes according to their rank; a node with rank r receives a link with probability $p(r) \sim r^{-\delta}$. Second, with probability ρ, each node is *reranked*, i.e. moved to a new position toward the front of the list. The new position i is chosen randomly with uniform distribution between 1 (the top position) and the node's current rank j, thus focusing on positive bursts (see Figure 6.9 for pseudocode). The node previously occupying position i is moved back to $i+1$, and so on. Simulations of this model

```
Given real δ, ρ and ranking function r(),
    desired number of nodes N
for t in 0 .. N do
    # Growth step
   Create new node t
   Assign links from t to existing nodes k,
       with P(k) ∼ r(k)^−δ.
    # Reranking step
    for each k (r(k) = j), with probability ρ,
       choose random i < j and set r(k) = i
       for r(ℓ) in i .. j do
          set r(ℓ) = r(ℓ) + 1
       end
    end
end
```

FIGURE 6.9. Pseudocode for the rank-shift model. In the case of traffic, instead of assigning links to existing nodes, we simply increment their traffic counts.

were performed using the empirical number of nodes N (cf. Table 6.1), and various values of the parameters ρ and δ. The effect of varying these parameters is discussed next.

6.4.3. Evaluation of Model

For $\rho = 0$ we recover the original ranking model, and the distribution of $\Delta x / x$ (for instance, the traffic or indegree of nodes) matches that of preferential attachment (cf. Figure 6.6(a)). This describes the behavior of the many topics that do not undergo sudden, large bursts of attention. These dynamics are reflected in the lognormal portion of the burst magnitude distributions (cf. Figure 6.2).

For $\rho > 0$ numerical simulations show that the tail of the popularity burst magnitude distribution shifts from a lognormal to a power law while the popularity distribution remains a power law; its exponent remains $\gamma = 1 + 1/\delta$, with an exponential cutoff now depending on ρ. This modification allows the model to capture the dynamics of topics undergoing large bursts of attention.

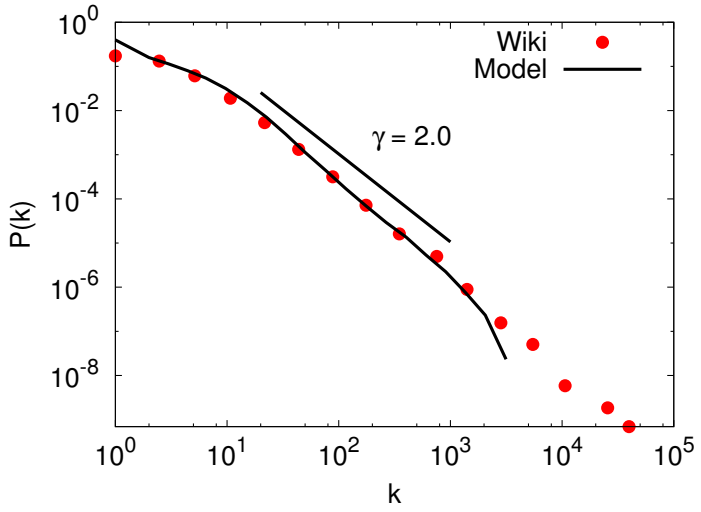


FIGURE 6.10. Agreement between the empirical indegree distribution from the December 2003 Wikipedia and the popularity distribution produced by the model. Both curves are consistent with a power law $P(k) \sim k^{-\gamma}$ with $\gamma \approx 2.0 \pm 0.2$. Results for traffic are similar.

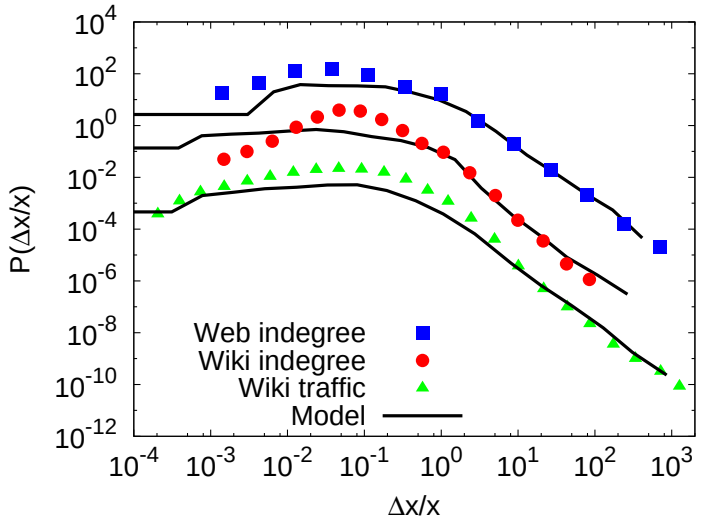


FIGURE 6.11. Agreement between the empirical popularity burst distributions and those produced by the model (data from 2003 for Chilean Web indegree, December 2003 for Wikipedia indegree, and a week in February 2008 for Wikipedia traffic). The curves are shifted for illustration purposes.

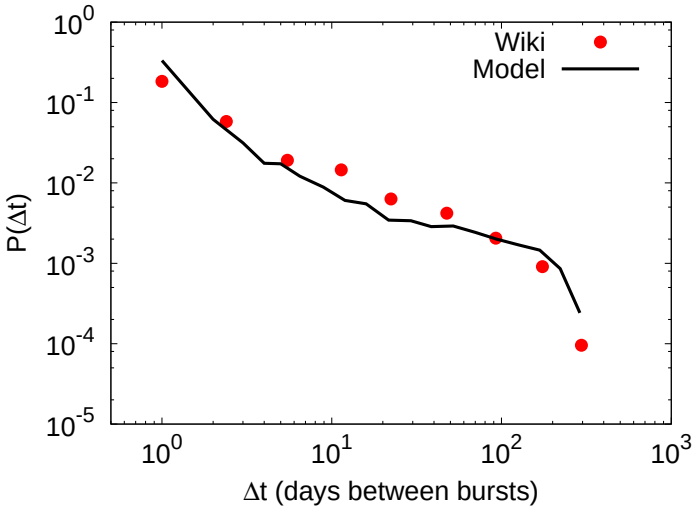


FIGURE 6.12. Agreement between the Wikipedia inter-burst time distribution and that produced by the model (data from the entire year 2003).

This behavior is manifest empirically in the broad tails of the burst magnitude distributions, which cannot be explained by preferential attachment alone (cf. Figure 6.6(a)).

Given the relationship between δ and the exponent γ of the indegree distribution (discussed above), we chose $\delta = 1/(\gamma - 1)$ using the empirical γ, finding $1 \leq \delta \leq 1.2$ for my data. We then numerically estimated a value of ρ in order to fit the distribution of $\Delta x/x$, and found $10^{-5} \leq \rho \leq 10^{-3}$. With these parameters, our simple model is able to reproduce many of the critical features observed in the empirical data. Not only does it predict the distributions of both popularity measures for both data sets (Figure 6.10), but also the long tail of the distributions of indegree and traffic burst size (Figure 6.11). Further, the model also captures the long-range distribution of inter-burst intervals (Figure 6.12). The rank-shift mechanism is therefore able to capture the way in which Web sites and pages gain and accumulate popularity: not by a gradual proportional process, but by a sequence of bursts that move them to the forefront of people's attention. This is sufficient to reproduce the broad distributions in the magnitude of bursts and in their temporal dynamics.

CHAPTER 7

POLITICAL DISCOURSE

7.1. Introduction

Up to this point, the systems I have examined have had the general flavor of a hyperlinked collection of pages. People navigated through this web of pages, with the number of links to a page or the number of users visiting it being considered its popularity. In this chapter, I outline an exploration of a different, but related, notion of popularity: the popularity of ideas themselves. The work here is based on a large corpus of data collected from Twitter. Using this data, and with appropriate definitions of what constitutes an idea (or *meme*), I am able to associate a meme's popularity with the number of users that are discussing it in their Twitter posts. On Wikipedia we can consider an idea to be synonymous with the page that discusses it. For instance, we might associate the popularity of 'Biology' as a general topic with the number of hits to the Wikipedia article on 'Biology.' With Twitter data, however, we can do more — not only can we track what ideas are being discussed, we can see *by whom* they are being discussed, and track the patterns of interactions between users when they are discussing particular ideas. We can also track the spread of ideas as they pass from user to user.

Here, I frame this discussion in the context of American politics, an area in which social media systems are becoming increasingly important [**Ben06, Sun07, FD08, AFL$^+$10**]. It has been repeatedly observed that people of similar beliefs and interests will tend to associate with each

other preferentially, compared to those different than them [**AG05**] (as well as visualizations like `http://www.orgnet.com/divided.html`). In this chapter I quantify this preference in the context of Twitter users discussing American politics, and find that it is stronger for some forms of communication than for others. In the process of doing this I outline methods for identifying tweets which are related to politics, and building the diffusion networks based on them.

7.2. Overview

I focus on data collected from the Twitter *gardenhose* between September 14th and November 1st, 2010 — the run-up to the November 4th American congressional midterm elections. This data set consists of about 354.5 million tweets. For a general overview of this dataset and Twitter, consult Chapter 4 § 4.6. To focus on data about a specific topic, I used some filtering techniques (outlined in the next section) to select just those tweets which concerned American politics. This left me with approximately 252,200 tweets involving around 45,000 users; it is on that corpus of tweets that the experiments discussed in this chapter are based. The original tweet dataset is available courtesy of Bruno Gonçalves [**GPV11**].

7.2.1. Identifying political tweets

While there exist many methods for identifying the topic of a document in general, Twitter's small document size of 140 characters precludes complicated statistical methods. Further, the wide adoption of hashtags on Twitter provides a convenient alternative to such complicated solutions. I thus consider a tweet to be politically relevant if it contains at least one political hashtag, reducing the problem of identifying political tweets to the simpler problem of identifying political hashtags. At this point, one approach would be to simply construct a list of a large number of such hashtags; indeed, this is the approach adopted in the work described in Chapter 8. However, I elected to use a more automatic method of building this set of political hashtags to avoid introducing bias.

I began by seeding the set of hashtags with the two most popular hashtags readily identifiable as political. These are the hashtags `#tcot` and `#p2`, being the self-adopted labels of the Top Conservatives on Twitter and Progressives 2.0, respectively. By choosing one hashtag from each side of the political debate, I hoped to give equal opportunity to left- and right-leaning users to be included in the final sample. I then formed two ranked lists of hashtags, one for each of the seed

TABLE 7.1. Hashtags co-occurring with #p2, #tcot, or both. Tweets containing any of these hashtags were included in the sample.

Just #p2	Both	Just #tcot
#casen #dadt	#cspj #dem #dems	#912 #ampat #ftrs
#dc10210 #democrats	#desen #gop #hcr	#glennbeck #hhrs
#du1 #fem2 #gotv	#nvsen #obama #ocra	#iamthemob #ma04
#kysen #lgf	#p2 #p21 #phnm	#mapoli #palin
#ofa #onenation	#politics #sgp	#palin12 #spwbt
#p2b #pledge	#tcot #teaparty	#tsot #tweetcongress
#rebelleft #truthout	#tlot #topprog #tpp	#ucot #wethepeople
#vote #vote2010	#twisters #votedem	
#whyimvotingdemocrat		
#youcut		

hashtags, where each hashtag's position in the ranked list was based on the Jaccard coefficient between it and the seed. Recall that the Jaccard coefficient is a measure of overlap between sets; thus, this ranks higher the hashtags that co-occur most often with each of the seed hashtags. Given this ranking, I picked a cutoff that resulted in a set with relatively high precision. The cutoff I chose, 0.005, isolated 66 hashtags; of these, I excluded 11 for their ambiguity. Tables 7.1 and 7.2 contain the included and excluded tags, respectively.

Note the high degree of overlap between the tags related to each seed, shown in the middle column of Table 7.1. This suggests that users from both sides of the political spectrum use many of the same hashtags. However, one can imagine that when, for example, left-leaning users use the hashtag #gop it may be for very different reasons than when the same hashtag is used by a person on the political right.

7.3. Analysis

7.3.1. Political Communication Networks

Having identified a set of politically-relevant tweets, I then construct networks representing political communication among Twitter users. I construct a network for each of the two modes of

TABLE 7.2. Hashtags which would otherwise have been included in the lists in Table 7.1, but which were excluded due to ambiguous or overly-broad meanings.

Excluded from #p2	Excluded from both	Excluded from #tcot
#economy #gay #glbt	#israel #rs	#news #qsn
#us #wc #lgbt		#politicalhumor

TABLE 7.3. Number of tweets, number of nodes (users), mention edges, and retweet edges for networks constructed from the set of tweets containing hashtags associated with #tcot, #p2, as well as the union of the two.

Network	**Tweets**	**Nodes**	**Mentions**	**Retweets**
#p2	242,516	44,414	15,596	62,643
#tcot	225,336	33,433	15,814	54,886
union	252,200	45,365	17,752	64,423

user-to-user communication possible on Twitter — retweets and mentions. For the retweet network I draw an edge running from user A to user B if B retweets content originally broadcast by user A. The mention network is defined similarly, with an edge running from user A to user B if A mentions B in a tweet (which causes that tweet to appear on B's home screen). Note that in each of these cases, we can imagine that a unit of information has flowed along the direction of the edge. Table 7.3 contains the number of tweets in the sets associated with #p2 and #tcot, as well as the number of nodes, mention edges, and retweet edges in their associated graphs. For comparison, the same values are shown for the set of tweets constructed by forming the union of the sets associated with #p2 and #tcot. This union set is hardly larger than either of its two components, indicating a high degree of overlap between them. Thus, in the following I forgo separate discussion of the networks associated with #p2 and #tcot separately, choosing instead to focus on their union.

In total, the union retweet network consists of 23,766 non-singleton nodes, with 18,470 nodes in its largest connected component (and 102 nodes in the next-largest). The union mention network is smaller, consisting of 10,142 non-singleton nodes with 7,175 nodes in its largest connected component (and 119 in the next-largest). Figure 7.1 displays the degree distributions of these networks; they display the kind of broad behavior that is expected.

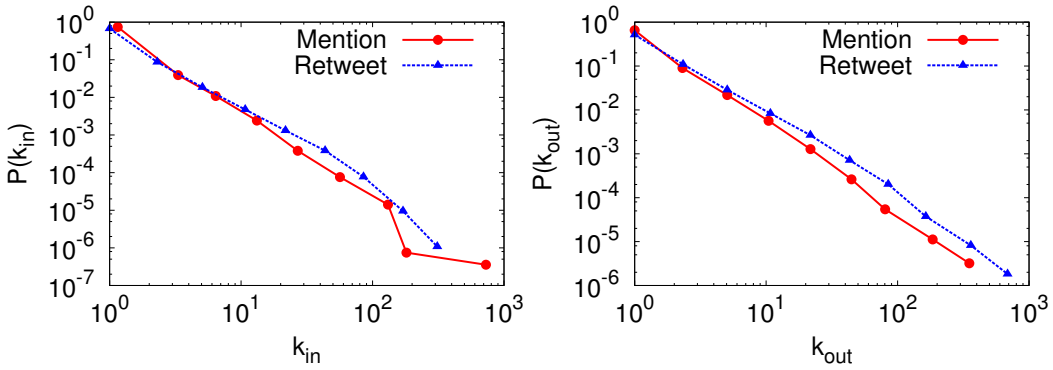


FIGURE 7.1. Distributions of in-degree (left) and out-degree (right) for the *union* mention and retweet networks.

7.3.2. Community Structure

Initial inspection of the network suggested that users might be significantly more likely to retweet users with whom they agree politically. I take a first step towards testing this hypothesis in this section, by examining whether users preferentially retweet other users, and whether users who retweet each other form clusters in the graph, as well as testing dual notions for the mention network. In work with Conover and others [**CRF$^+$11**] we expand on this analysis, examining retweeting users to determine political alignment.

In the following, I consider the retweet and reply networks to be synonymous with their largest connected components.

I perform community detection using a label propagation method [**RAK07**], restricted to two cluster labels. Starting with an initial arbitrary label (i.e. a cluster membership) for each node, this method works by iteratively assigning to each node the label that is shared by most of its neighbors. Ties are broken randomly when they occur. This is a greedy algorithm and can easily converge to local optima. Thus, rather than assigning the initial labels randomly, I use assignments produced by Newman's leading-eigenvector modularity maximization method for two clusters [**New06a**]. I further note that the label propagation method can return different assignments in subsequent runs for the same graph and the same initial conditions, due to the randomness involved in breaking ties. Thus, to check that my choice of seeds would generate similar partitionings in subsequent runs of the algorithm, I ran it 100 times for each of the mention and retweet networks and compared the results. Table 7.4 reports the high average agreement between the resulting 4,950 unique pairs of

TABLE 7.4. Minimum, maximum, and average ARI similarities (cf. 2.2.3) between 4,950 unique pairs of cluster assignments computed by label propagation for each of the mention and retweet networks.

Graph	Min	Max	Mean
Mention	0.80	1.0	0.89
Retweet	0.94	0.98	0.96

cluster assignments for each graph, as computed by the Adjusted Rand Index [**HA85**]. This high average agreement suggests that I need only run the label propagation algorithm one time, and may avoid any kind of consensus clustering for simplicity's sake.

I thus compute cluster assignments by seeding the label propagation algorithm with clusters from Newman's leading eigenvector method. By this method, the final cluster assignment for the retweet and mention networks resulted in modularities of 0.48 and 0.17, respectively (see 2.2.1 for the definition of modularity). Figure 7.2 shows the retweet and mention networks, laid out using a force-directed layout algorithm [**FR91**], and with node colors and shapes determined by the node's assigned community. Note that the retweet network exhibits two distinct communities of users, while the mention network is dominated by a single cluster of interconnected users.

While Figure 7.2 and the modularity values might suggest to us that the retweet network is more amenable to being split into two clusters than is the mention network, I cannot compare the two modularities directly as the networks are of different sizes. I thus need a way to compare the 'goodness' of cluster assignments across different graph sizes. I compute this 'cluster goodness' by creating some number N of random graphs with the same degree sequence as the original graph, clustering those graphs by the same method as used for the original graph, and comparing the modularity of the original partition on the original graph with the modularities of the partitions on the random graphs. These modularities can be viewed as observed values of a random variable X, being the modularity of a partition on a random graph. The intuition here is that the degree to which the modularity of the original graph is larger than those in the sampled values is a measure of how much more amenable to being split into two clusters the graph is than would be expected by chance. Further, since this measure avoids comparing the modularities of graphs of different sizes, I may use it to compare the 'clusterability' of the retweet graph with that of the mention

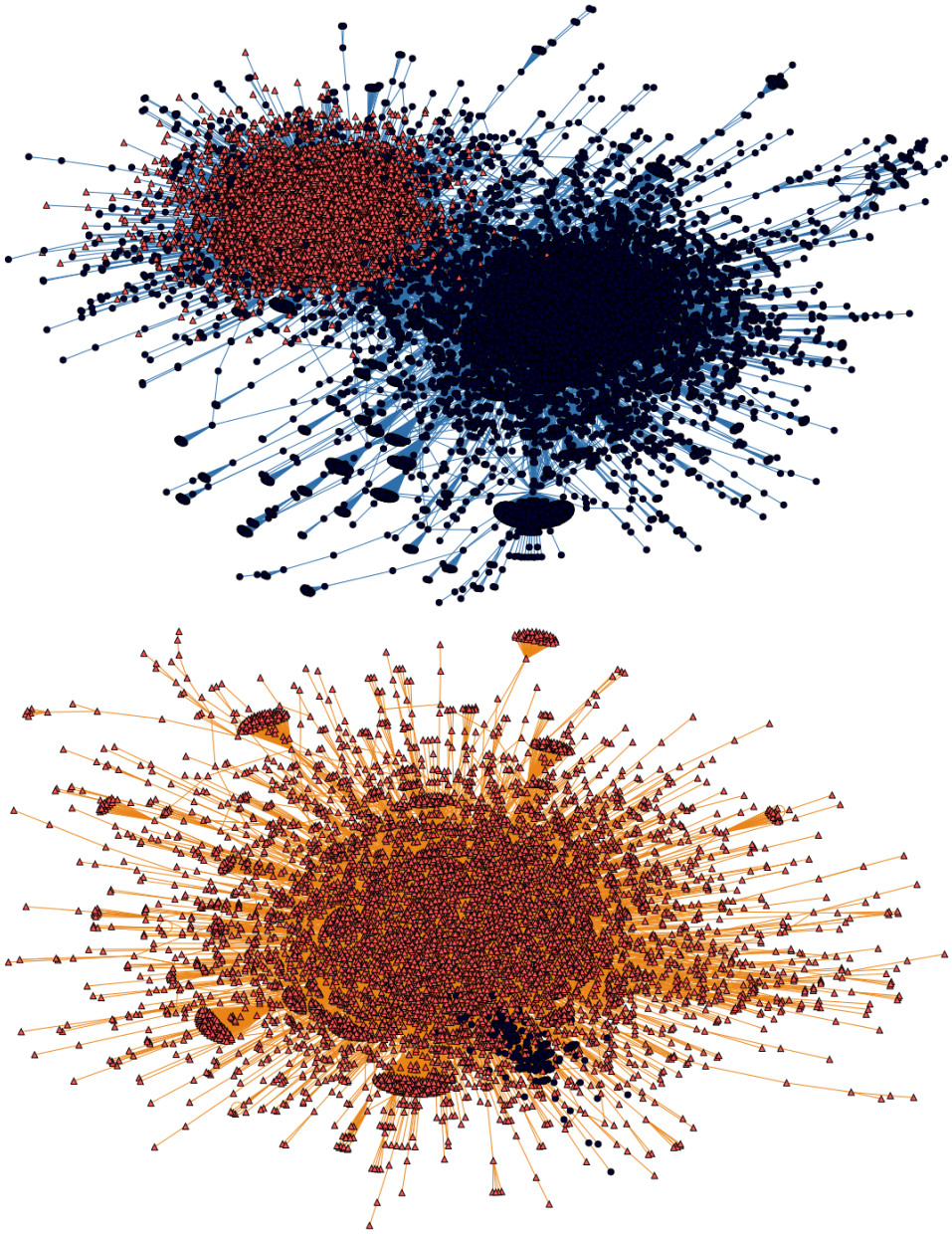


FIGURE 7.2. The *union* retweet graph (top) and mention graph (bottom), laid out using a force-directed layout algorithm. Node colors reflect cluster assignments, as computed using a label-propagation algorithm with each node initialized randomly to one of two clusters. Structure is readily apparent in the retweet network, but less so in the mention network.

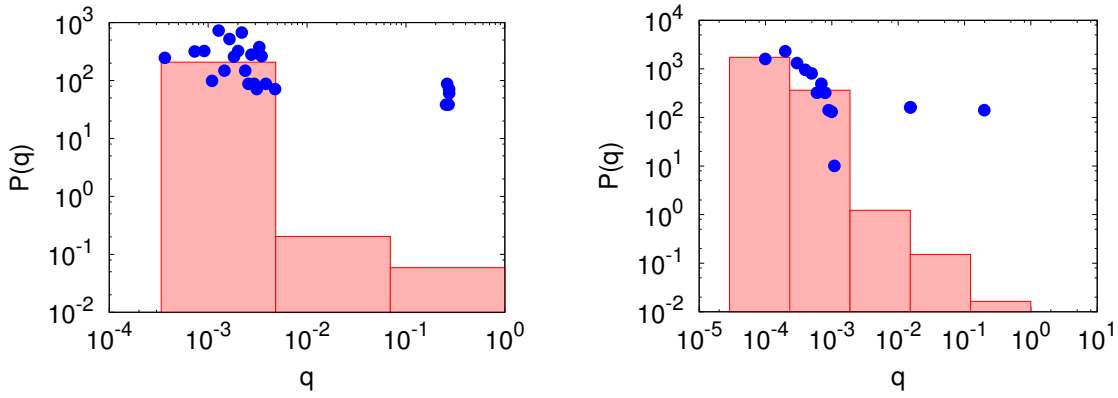


FIGURE 7.3. Distributions of modularities of random networks with the same degree sequence as the mention network (left) and retweet network (right). The boxes represent the log-width binning of the data, while the points are the actual modularity values.

graph. I thus construct 1,000 random graphs with the same degree sequence as the mention graph, and cluster each of them, resulting in 1,000 modularity values. I do the same for the retweet graph, yielding another 1,000 modularity values.

It is now necessary to compare the modularities of the original graphs with the randomly-sampled observations. This is complicated by the fact that these sampled modularities, shown in Figure 7.3, do not fall into a readily-identifiable distribution. There are thus two approaches that may be taken: I can use the peakedness of the distributions to argue that the few larger values are outliers, and that the data are approximately normal; I may then use the Z-scores of the modularity of each original network, with respect to its random samples, for comparison. I might also avoid the assumption of normality by using a more general measure. Each of these methods is presented in turn.

If we assume the distribution of the sampled modularity values is approximately normal, I can use them to compute the Z-score for the mention network's modularity; it is $Z_m = 2.06$. I contrast this with the Z-score for the retweet network's modularity, computed in a similar fashion, which is $Z_r = 11.02$. Thus, I can conclude that the bi-clustered structure found in the retweet network is far more significant than that found in the mention network.

If we would rather not make the assumption of normality, I can reach the same conclusion — namely that the retweet network is more well-clustered than is the reply network — by another

argument. I use Chebyshev's inequality,

$$\Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}, \tag{23}$$

which allows me to place a conservative bound on the probability that the random variable X, being the modularity of a sampled graph, will take on the value of the original graph's modularity. Solving the above for k, and using the observation that $X > \mu$ in my case, I have

$$\Pr(|X - \mu| \geq k\sigma) = \Pr\left(\frac{X - \mu}{\sigma} = Z \geq k\right) \leq \frac{1}{k^2} \tag{24}$$

I can therefore use $k_r = Z_r$ and $k_m = Z_m$ for the Z_r, Z_m computed previously to find

$$\Pr(Z \geq Z_m) \leq \frac{1}{Z_m^2} = 0.24, \tag{25}$$

$$\Pr(Z \geq Z_r) \leq \frac{1}{Z_r^2} = 0.008, \tag{26}$$

Thus, since a network that can be clustered as well as the retweet network is much less likely to arise randomly (relative to the mention network), I can conclude based on this method as well that the clustering in the retweet network is much more pronounced.

From all this, I can conclude that the users have a very different preference for whom they retweet than they do for mentioning. People seem to self-segregate very well into strong communities in the retweet network, but very little of such structure is present in the mention network.

7.3.3. Content analysis

The node clustering explored in the previous section was accomplished solely based on network properties of the users involved; does it have any significance in terms of the actual *content* of their discussions? As a first step towards answering this question, I consider each user to be associated with a pseudo-document containing all the hash tags in their tweets. I can then compute the cosine similarities between each pair of user documents, separately for users in the same cluster and users in different clusters. Figure 7.4 shows the distributions of these similarities. The data for the *retweet* network show that users placed in the same cluster are markedly more likely to be similar to each other, in contrast with when clustering is performed on the *mention* network. Further, both in the mention and retweet networks, it is the case that one of the clusters is more cohesive than the other — its users are more strongly related to each other, on average.

An important second step in this analysis is to interpret the clusters in terms of the political affiliations of their members, to see if one mainly consists of left-leaning users while the other contains right-leaning users. This analysis is not discussed here, but it shoes that this is indeed the case — cluster assignments exhibit an accuracy of greater than 90% in predicting political alignment, with the latter determined by human inspection of the tweets by each user [**CRF$^+$11**].

7.4. Tag use and user mentions

So far in this chapter, several related observations have emerged. Firstly, there is a high degree of overlap between tweets associated with `#p2` and `#tcot`. This degree of overlap is surprising if one considers these tags to be strongly associated with the left and right, respectively. Secondly, while strong segregation is present in the retweet network (which is shown elsewhere to correspond strongly to political alignment [**CRF$^+$11**]), there is little segregation in the mention network. These observations suggest two related hypotheses:

- There is a higher degree of connectivity in the mention network between users who 'disagree' (in terms of their political alignment) than there is in the retweet network.
- There is, practically speaking, no such thing as a hashtag that 'belongs' to the left or the right. Many hashtags, if at all popular, are used frequently by both types of users. However, the *proportions* of right- and left-leaning people using a particular hashtag may vary.

These hypotheses give rise to a third:

- When a user of a certain political alignment (say left) uses a hashtag that is very often used by members of the opposite alignment (e.g. right), this may provoke members of that alignment to respond to the original poster via mentions. Thus, there is a correlation between the frequency of such use, and the frequency of connections (via mentions) to users of the opposite alignment.

In testing these hypotheses, I use a corpus of 1,000 labeled users made available by Conover and Francisco [**CRF$^+$11**]. These users were chosen uniformly at random from the users present in both of the mention and retweet networks mentioned in this chapter, and then assigned labels by a human reviewer. Users could be labeled as 'left,' 'right,' or 'undecidable.' In the following I ignore the 'undecidable' category and focus simply on the 'left' and 'right' labels.

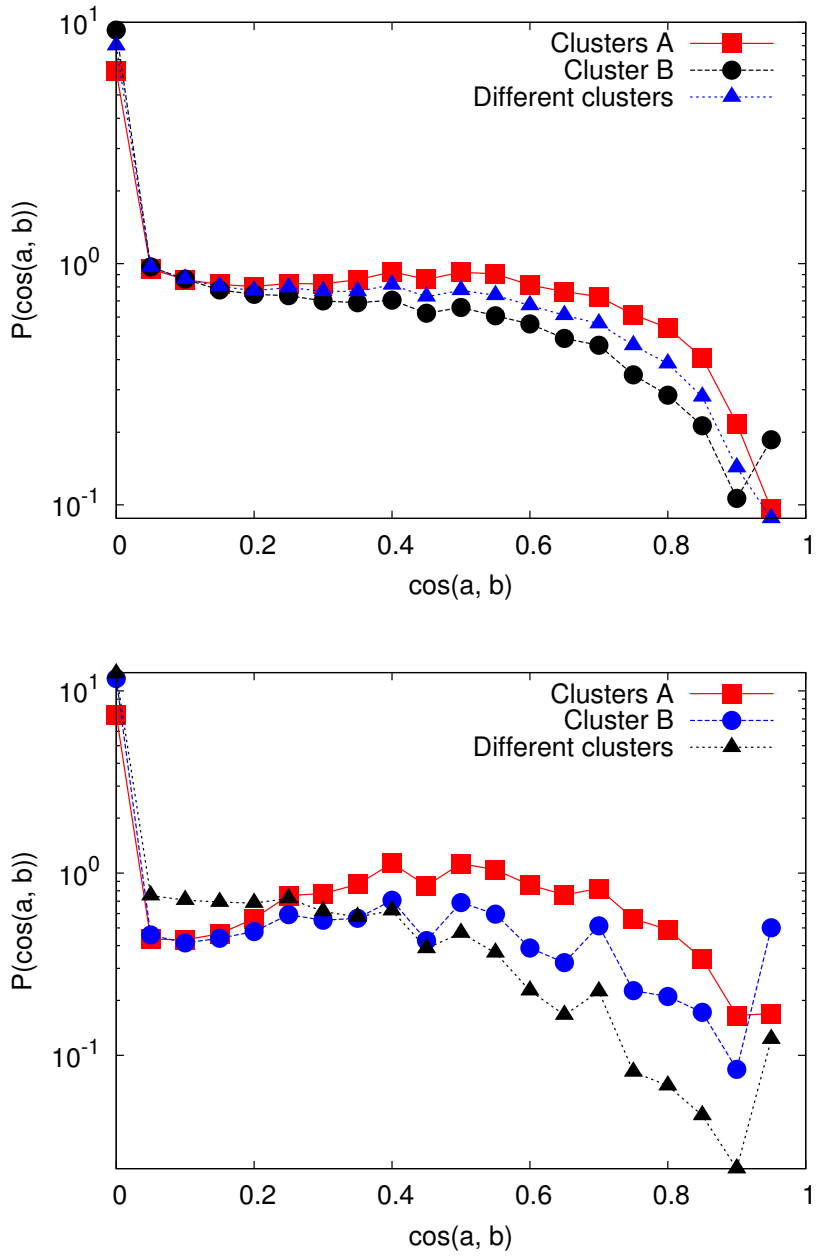


FIGURE 7.4. Distribution of the cosine similarity between pairs of users for the mention network (top) and retweet network (bottom). Note that in the retweet network, the users in the same cluster are more similar than users in different clusters. This is not true in the mention network, which shows higher similarity between clusters than within cluster B.

TABLE 7.5. Ratio between observed and expected number of links between users of different political alignments. Values for the Mention (left) and Retweet (right) networks are shown.

	$\mapsto$ **Left**	$\mapsto$ **Right**	$\mapsto$ **Left**	$\mapsto$ **Right**
Left	1.23	0.68	1.70	0.05
Right	0.77	1.31	0.03	2.32
	Mention		Retweet	

7.4.1. How likely are users to mention those with whom they disagree?

Here I explore the number of 'mention' connections between users who disagree (in terms of their human-assigned labels), compared with the relative number of retweet connections between disagreeing users. To do this, I analytically compute the expected value of the number of such links, in a network where the destination edges of all links were detached from their original positions and reattached uniformly at random. Let the number of users labeled as 'left' and 'right' be U_L and U_R, respectively. Let the number of edges originating from users labeled 'left' be k_L, and define k_R analogously. Then the expected number of edges from left-leaning to right-leaning users is simply the number of edges originating from the left times the fraction of all users that are right-leaning. That is,

$$\mathrm{E}\left[R \to L\right] = k_R \cdot \frac{U_L}{U_L + U_R} \tag{27}$$

I can compute the other expected numbers of edges ($R \to R$, $L \to R$, $L \to L$) in the analogous way. In Table 7.5 I report the ratio between these expected numbers of links and those observed in the data. We see that for both means of communication, users are more likely to engage other users with whom they agree. However, this effect is much more pronounced in the mention network.

7.4.2. Are hashtags 'left' or 'right?'

Earlier I observed that it does not seem to be the case that popular hashtags are exclusively used by members of one political alignment or the other. To explore this phenomenon in more detail, I use the notion of *political valence*, a measure that encodes the relative frequency that a tag is used by left- and right-leaning users [**CRF$^+$11**]. Let T be the set of all tags, and let $N(t, L)$ and $N(t, R)$

TABLE 7.6. Valences of the top 20 tags, by popularity. Note that #tcot and #p2 are strongly right and left, respectively, but not overwhelmingly so.

Tag	Count	Valence	Tag	Count	Valence
#tcot	17676	0.39	#hhrs	1027	1.00
#p2	10446	-0.60	#cspj	888	0.98
#teaparty	6910	0.35	#desen	864	0.34
#tlot	2988	0.19	#p21	860	-0.81
#gop	2898	0.12	#news	800	0.53
#sgp	2887	0.71	#tpp	662	0.18
#ocra	2207	0.34	#obama	656	0.12
#dems	1204	-0.81	#mapoli	564	-0.13
#twisters	1109	0.84	#hcr	555	-0.57
#palin	1052	0.36	#nvsen	550	-0.11

be the frequency with which a tag $t \in T$ is used by left- and right-leaning users, respectively. For notational convenience, let $N(R) = \sum_t N(t, R)$ be the total number of hashtags used by right-leaning users, and define $N(L)$ similarly. The valence of t is then defined by

$$\mathrm{V}(T) = 2 \times \left(\frac{N(t,R)/N(R)}{N(t,R)/N(R) + N(t,L)/N(L)} \right) - 1 \tag{28}$$

The translation and scaling constants serve to bound the measure between -1 for a tag only used by the left, and $+1$ for a tag only used by the right. Table 7.6 shows the valences of the top 20 hashtags, ranked by number of appearances. Some interesting patterns are present; the flagship conservative and liberal hashtags, #tcot and #p2, have valences that suggest that while they are mostly used by their side's supporters, the other side chimes in too. In contrast, topics of general interest and neutral affect like #obama have valences close to 0, as they are discussed by both sides. Figure 7.5 displays the mean valence of tags as a function of the number of times they were used, as well as the numbers of hash tags used to compute each mean. Note that it is not the case that popular tags often have neutral valence. Rather, the broad distribution at all levels of use suggests that any hashtag used a non-trivial number of times will have a valence that is less than one in absolute value. Of course, hashtags used only one time must always have a valence of either 1 or -1.

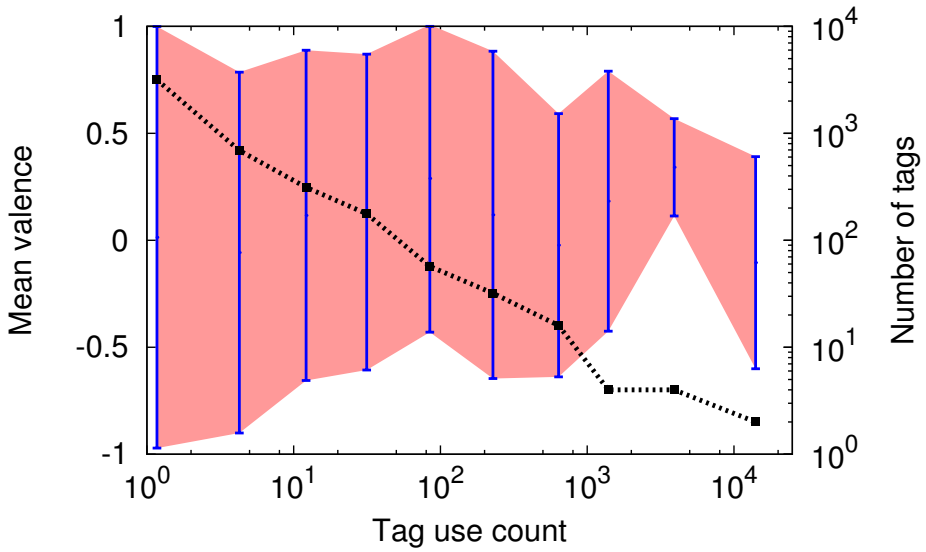


FIGURE 7.5. Correspondence between the number of times a tag was used and the mean valence of tags used that number of times. It is not the case that popular tags have valences close to zero. The error bars are given for one standard deviation from the mean. The overlaid dashed line corresponding to the right y-axis indicates the number of actual hashtags from which the mean valences are computed.

7.4.3. Does a mix of hashtags promote linking?

Given the above measure of hashtag valence, we can define the general valence of a user as the mean valence of all the hashtags that they use in posts. Thus a user who uses a mix of hashtags associated with each side will have a valence closer to zero. These users might not be necessarily more moderate voices, but people who want to engage the other side of the debate. A hypothesis I can then test is whether they do engage the other side more than do users who have a larger absolute valence. I test this hypothesis by computing the mean tag valence for all users among those who appear in both the retweet and mention networks. We already know that the number of retweet links across clusters is very small, and the clustering of the mention network is not meaningful. Therefore, I use the cluster assignment from the *retweet* network, and count the number of times that a user in retweet cluster A *mentions* a user in retweet cluster B. Figure 7.6 shows the mean number of such links as a function of the mean user valence. The peak around low absolute

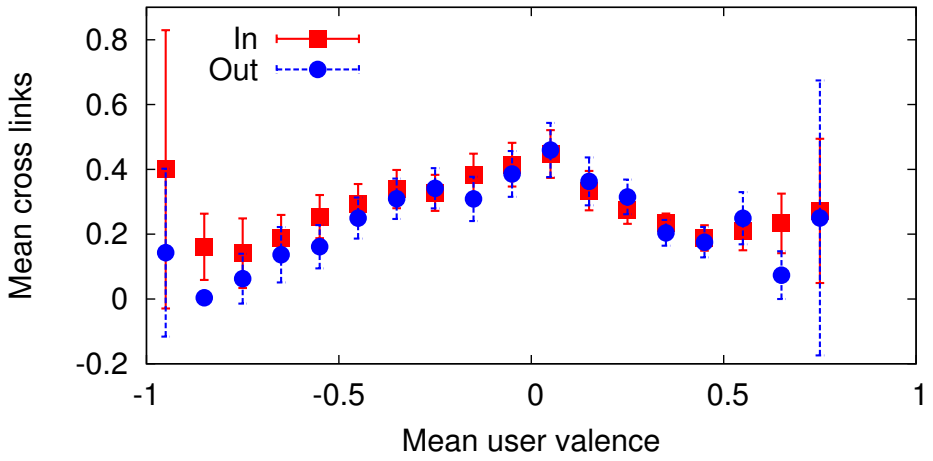


FIGURE 7.6. The average number of mention-links from the opposing retweet-cluster that a user receives and produces, as a function of their mean valence. The peak around zero suggests that users who use tags of both positive and negative valence are more likely to receive and produce links.

values of valence suggests that indeed, users who use a mix of tags across the valence spectrum, or moderate tags, are more likely both to create and receive inter-cluster links.

7.5. Conclusion

This chapter has described the construction of a corpus of 250,000 political tweets, and the construction of interaction networks from this corpus. These networks, based on the two major modes of inter-user communication in Twitter, can then be analyzed to determine the patterns of connections between users in relation to those users' political beliefs. I describe the clustering of the networks using a combination of off-the-shelf clustering methods — label propagation for the final clustering, seeded by Newman's leading eigenvector method to avoid the instability caused by random seeds. In the retweet network, these clusters correspond very well with the political ideologies of the users inside them — conservatives and liberals are very often clustered together. The mention network, however, shows much less of this preferential behavior. I also outline some experiments designed to explore the link between these connection patterns and the messages produced by the users involved. Research of this type has the potential to identify the 'moderate

voices' in a discussion space, as well as the people who speak only with others with whom they agree.

This is an exciting and largely unexplored area in which much further work is possible. Conover *et al.* apply machine learning techniques to determine a user's political leaning [**CRF$^+$11**]. None of the features and techniques they explore significantly outperform using the clusters described above as a single feature. Further investigation is also needed to explore the relationship between the mention network and the political affiliations of the users it links. In the next chapter I explore some applications of analysis in this political speech, focusing on tracking the diffusion of political memes, and on identifying content propagated in a deceitful way.

CHAPTER 8

TRUTHY: A CASE STUDY

8.1. Introduction

In this chapter, I combine techniques and datasets from previous chapters in the production of a system designed to detect political *astroturf* on Twitter. *Astroturf* refers to messages which are deceitfully propagated so as to appear similar to real 'grassroots' campaigns; the name refers to AstroTurf, a brand of imitation grass used in sports stadiums.

The experiments described here were inspired by some work done by Metaxas and Mustafaraj, studying a political smear campaign during the 2010 Massachusetts special election. Their paper describes a concerted, deceitful attempt to cause a certain URL to rise to prominence on Twitter, and to make it appear that the URL was spread by a groundswell movement [**MM10**]. The success of the attack was sufficient to lead to subsequent viral spread of the URL by legitimate means. Originally propagated by a network of nine collaborating accounts, the URL was spread enough that appeared on the front page of a Google search for the political candidate it mentioned. This type of attack, which can be mounted very cheaply and could potentially reach an even larger audience than traditional advertisements, will certainly be used again.

While some of the techniques associated with spam (such as the mass creation of accounts meant to look like real users) are shared with political astroturfing, spam and astroturf differ in several ways. Spammers are often interested in causing users to click a link; in contrast, astroturfers

want a particular tweet or idea to have a false sense of group consensus. Further, many of the users involved in propagating a successfully astroturfed message may in fact be legitimate users, unwittingly complicit in the deception, having been themselves deceived by the original core of automated accounts. Thus, astroturf detection methods cannot only rely solely a message's content, or on features of the accounts of the users who propagate it.

In this chapter, I describe some of my contributions to a system designed to detect political astroturf and its associated Web site, which we collectively call 'Truthy.' This term, which we also use to refer to the political astroturf that the system detects, is borrowed from the comedian Stephen Colbert. It is used to describe something that a person claims to know based on emotion rather than evidence or facts. The Truthy system is the work of many others besides myself, so I describe below only the components which I contributed. The system is available online at `truthy.indiana.edu`; a more thorough treatment of the system is given in a recent paper [**RCM+10**].

I begin the chapter by defining the units of information that Truthy tracks, and how the diffusion networks for each of these units is built. I then describe the technical details of the systems built to support this tracking. Finally, I outline some promising results for the automatic classification of truthy memes.

8.2. Meme Types

Before beginning discussion of the Truthy system itself, I first define the fundamental units of information that Truthy tracks, referred to as *memes.* While in general a meme could be any abstract topic, we forgo sophisticated topic modeling techniques and focus on features unique to Twitter data which can be used as topic markers — *hashtags* and *mentions*. Hashtags are tokens, included in the text of a tweet and prefixed by a hash (#), that are used to label the topical content of tweets. Some examples of popular tags are `#gop` and `#obama`, marking discussion about the Republican party and President Obama, respectively. A Twitter user can call another user's attention to a particular post by including that user's screen name in the post, prepended by the `@` symbol. These *mentions* can be used as a way to carry on conversations between users, or to denote that a particular Twitter user is being discussed. Besides hashtags and mentions, Truthy also tracks URLs, as well as the text of each tweet when all URLs and metadata markup have been removed. Information

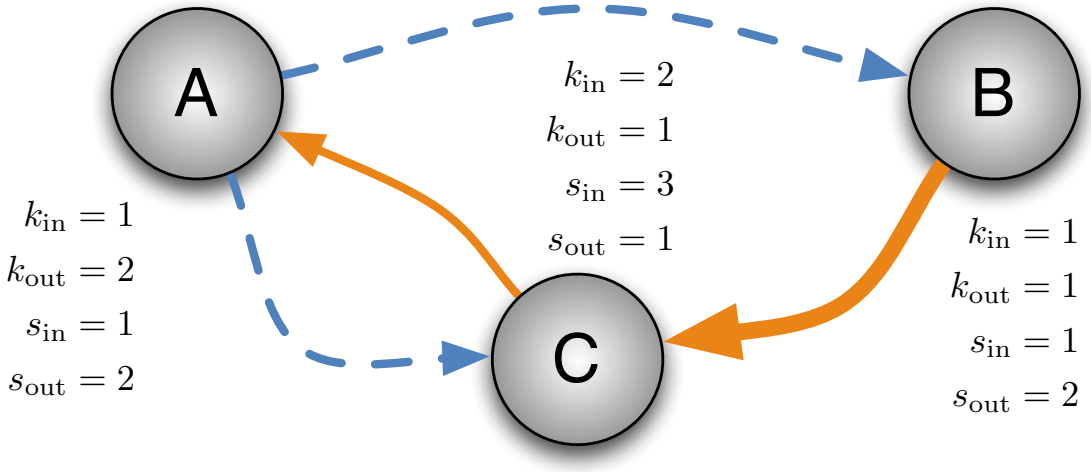


FIGURE 8.1. Example of a meme diffusion network involving three users mentioning and retweeting each other. The values of various node statistics are shown next to each node. The strength s refers to weighted degree.

about the propagation of each of these types of memes is used to build networks representing the diffusion of information among users.

8.2.1. Network Edges

To represent the flow of information through the Twitter community, I again construct a directed graph in which nodes are individual user accounts (as in Chapter 7). An example diffusion network involving three users is shown in Figure 8.1. An edge is drawn from node A to B when either B is observed to retweet a message from A, or A mentions B in a tweet. The weight of an edge is incremented each time we observe an event connecting two users. In this way, either type of edge can be understood to represent a flow of information from A to B. Observing a retweet at node B provides implicit confirmation that information from A appeared in B's Twitter feed, while a mention of B originating at node A explicitly confirms that A's message appeared in B's Twitter feed. This may or may not be noticed by B, therefore mention edges are less reliable indicators of information flow compared to retweet edges. In contrast with Chapter 7, the networks we build in Truthy are directed; also, they contain both types of edges (retweet and mention). We do not build separate networks for each of these edge types. The mechanism for building networks is

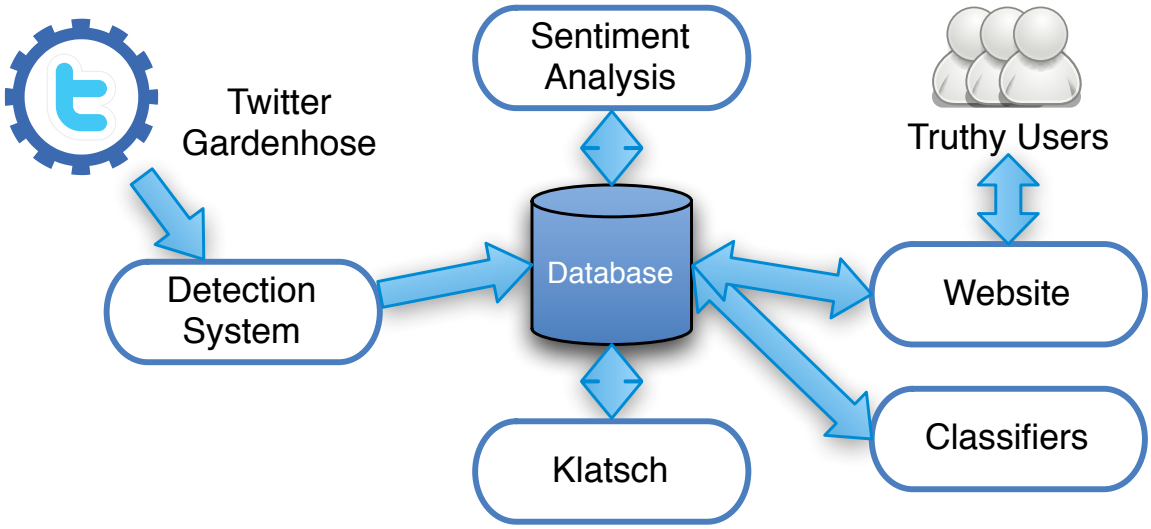


FIGURE 8.2. The Truthy system architecture.

mainly due to Mark Meiss and the Klatsch framework [**RCM**$^{+}$**10**]; I present it here to illustrate the information that must be extracted from the raw tweets.

We determine who was replied to or retweeted not by parsing the text of the tweet, which can be ambiguous (as in the case when a tweet is marked as being a 'retweet' of multiple people). Rather, we rely on Twitter metadata that we download along with the text of the tweet, and which designates users as being the users replied to or retweeted by each message. Thus, while the text of a tweet may contain several mentions, we only draw an edge to the user who is explicitly designated as the mentioned user by the tweet metadata. Note that this is separate from our use of mentions as memes, which we parse from the text of the tweet.

8.3. Truthy System Architecture

A general overview of the components of Truthy is shown in Figure 8.2. Truthy includes several components: a low-level system overseeing the collecting and processing of the raw data feeds from the Twitter API, the meme detection framework, The Klatsch framework responsible for computing key network statistics and layouts, and a Web based presentation framework that allows us to collect user input on which memes the community deems most suspicious. I describe here the

detection system, website, and classifiers; for more on the rest of the system, see [**RCM+10**]. Network statistics and community-generated annotations are the primary inputs to the classification apparatus discussed in § 8.5.

8.3.1. Streaming Data Collection

To collect meme diffusion data, Truthy relies on whitelisted access to the Twitter 'gardenhose.' The gardenhose provides detailed data on a sample of the Twitter corpus at a rate that varied between roughly 4 million tweets a day near the beginning of the study described here, to around 8 million tweets per day at the end. I distinguish here between the gardenhose and the firehose, the latter of which provides an unfiltered dump of all Twitter's traffic, but is only available to entities thats purchase access. While the process of sampling edges (tweets between users) from a network to investigate structural properties has been shown to produce suboptimal approximations of true network characteristics [**LF06**], I will show that the data extracted is still useful for several purposes.

All collected tweets are stored in files at a daily time resolution. We maintain files both in a verbose JSON format containing all the features provided by Twitter, and in a more compact format that contains only the features used in our analysis. This collection is accomplished by a component of Truthy that operates asynchronously from the others, and was implemented by Bruno Gonçalves. The detection and tracking steps, described next, are my work.

8.3.2. Meme Detection

A second component of Truthy is devoted to scanning the collected tweets in real time, by pulling data from the daily files described above. The task of this meme detection component (Figure 8.3) is to determine which of the collected tweets are to be stored in our database and subjected to further analysis. My goal here is to collect only tweets *a)* with content related to the political elections, and *b)* of sufficiently general interest. I implemented a filtering step for each of these criteria, described below.

8.3.2.1. Tweet filter. To identify politically relevant tweets, I turn to a hand-curated collection of approximately 2500 keywords relating to the 2010 U.S. midterm elections. This keyword list contains the names of all candidates running for U.S. federal office, as well as any common variations and known Twitter account usernames. The collection further contains the top 100 hashtags that

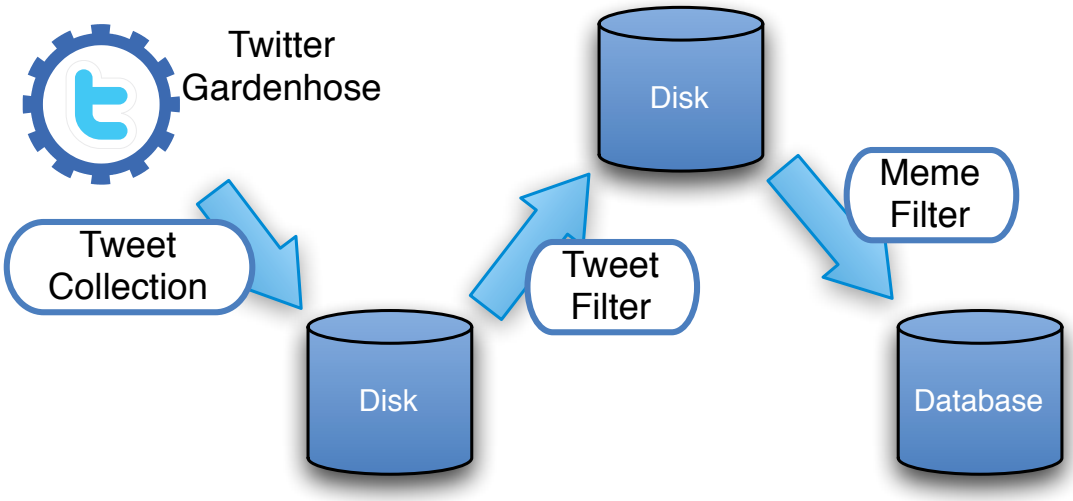


FIGURE 8.3. The meme detection and tracking system consists of three separate, asynchronous components — the tweet collection, which downloads tweets and saves them to disk; the tweet filter, which determines tweets likely to relate to politics; and the meme filter, which identifies memes of significant general interest and saves them in the database.

co-occurred with the hashtags `#tcot` and `#p2` (the top conservative and liberal tags, respectively) during the last ten days of August 2010. The motivation for including explicit hashtags in the filter is not to ensure that these terms are tracked by the system (though this is a side effect), but rather to capitalize on the common behavior of Twitter users whereby they include chains of tags to identify multiple relevant topics of interest. Thanks for this list of keywords is due to Eni Mustafaraj. This component, too, operates asynchronously. It is capable of processing tweets at a rate of about 10 times faster than our sampling rate, allowing it to easily handle bursts of traffic. I refer to this component as the `tweet_filter`.

8.3.2.2. Meme filter. Simply including all the tweets at this step would have resulted in a proliferation of distinct memes, as it would have included as a meme any hashtag, URL, username, or phrase mentioned by any user even one time. I thus implemented a second stage of filtering designed to identify those tweets containing memes of sufficiently general interest. I refer to this stage of filtering as the `meme_filter`.

```
Repeat:
   Get next tweet from priority queue
   Throw away tweets from the sliding activation
        window that are more than 1 hour old
       (where current time is determined from the
        just-fetched tweet)

   Track activation of all this tweet's memes

   For each activated meme (m) in this tweet:
      store all un-stored tweets seen in the
          past hour that are related to m
```

FIGURE 8.4. Pseudocode for the main loop of the `meme_filter`.

The `meme_filter`, like the `tweet_filter`, reads tweets in real time. However, since tweets in the gardenhose are not guaranteed to be in strict temporal order, the `meme_filter` inserts all tweets read into a priority queue that orders them by their timestamp. Tweets are then processed in the order that they are removed from the queue. This does not guarantee that tweets will be read in sorted order, but greatly decreases the number of out-of-order tweets — for a priority queue of size n, any tweet less than n places out of order will be correctly ordered. I found empirically that $n = 1000$ decreased out-of-order tweets to manageable levels of only a few per day (out of millions). It is necessary to present tweets in-order to subsequent layers, to make maintenance of the sliding activation window (described next) more efficient. Thus any out-of-order tweets remaining after this step are discarded.

The `meme_filter`'s goal is to extract only those tweets that pertain to memes of significant general interest. To this end, I extract all memes (of the types described in § 8.2) from each incoming tweet, and track the activation over the past hour of each meme, in real time. If any meme exceeds a rate threshold of five mentions in a given hour it is considered 'activated;' any tweets containing that meme are then stored. If a tweet contains a meme that is already considered activated due to its presence in previous tweets, it is stored immediately. When the mention rate of the meme drops

below the activation limit, it is no longer considered activated and tweets containing the meme are no longer automatically stored. Note that a tweet can contain more than one meme, and thus the activation of multiple memes can be triggered by the arrival of a single tweet. I chose a low rate threshold with the understanding that if a meme is observed five times in our sample it is likely mentioned many more times in Twitter at large. The general algorithm for the `meme_filter` is shown in Figure 8.4.

The tracking of a new tweet consists of three steps: *(i)* removing tweets outside the current sliding activation window; *(ii)* extracting memes from the tweet and tracking their activation; and *(iii)* storing tweets related to any now activated memes. Because the tweets are presented in sorted order, and the number of memes in a tweet is bounded by the constant tweet length, step *(i)* can be completed in time linear in the number of old tweets, and steps *(ii)* and *(iii)* require constant time.

Prior to settling on this detection strategy for topics of general interest, we experimented with a more complicated strategy based on examining the logarithmic derivative of the number of mentions of a particular meme, computed hourly. This approach was inspired by previous work on attention dynamics in Wikipedia [**RMF$^+$10**] (and Chapter 6). Since many memes with bursty behavior have low volume, I augmented the burst detection algorithm with a second predicate that included memes that appeared in a minimum percentage of the tweets over the past hour. We eventually discarded this hybrid detection mechanism due to the complexity of choosing appropriate parameters, in favor of the simpler scheme described above.

The Truthy system has tracked a total of approximately 305 million tweets collected from September 14 until October 27, 2010. Of these, 1.2 million contain one or more of our political keywords; detection of interesting memes further reduced this set to 600,000 tweets actually entered in our database for analysis.

8.3.3. Web Interface

Truthy also includes a dynamic Web interface to allow users to inspect memes through various views, and annotate those they consider to be truthy. Raw counts of these user annotations are used as input to the classification apparatus described in § 8.5. To facilitate the decision making process, we provide a mixed presentation of statistical information and interactive visualizations elements. Snapshots of summary and detailed views available on the Truthy site are shown in

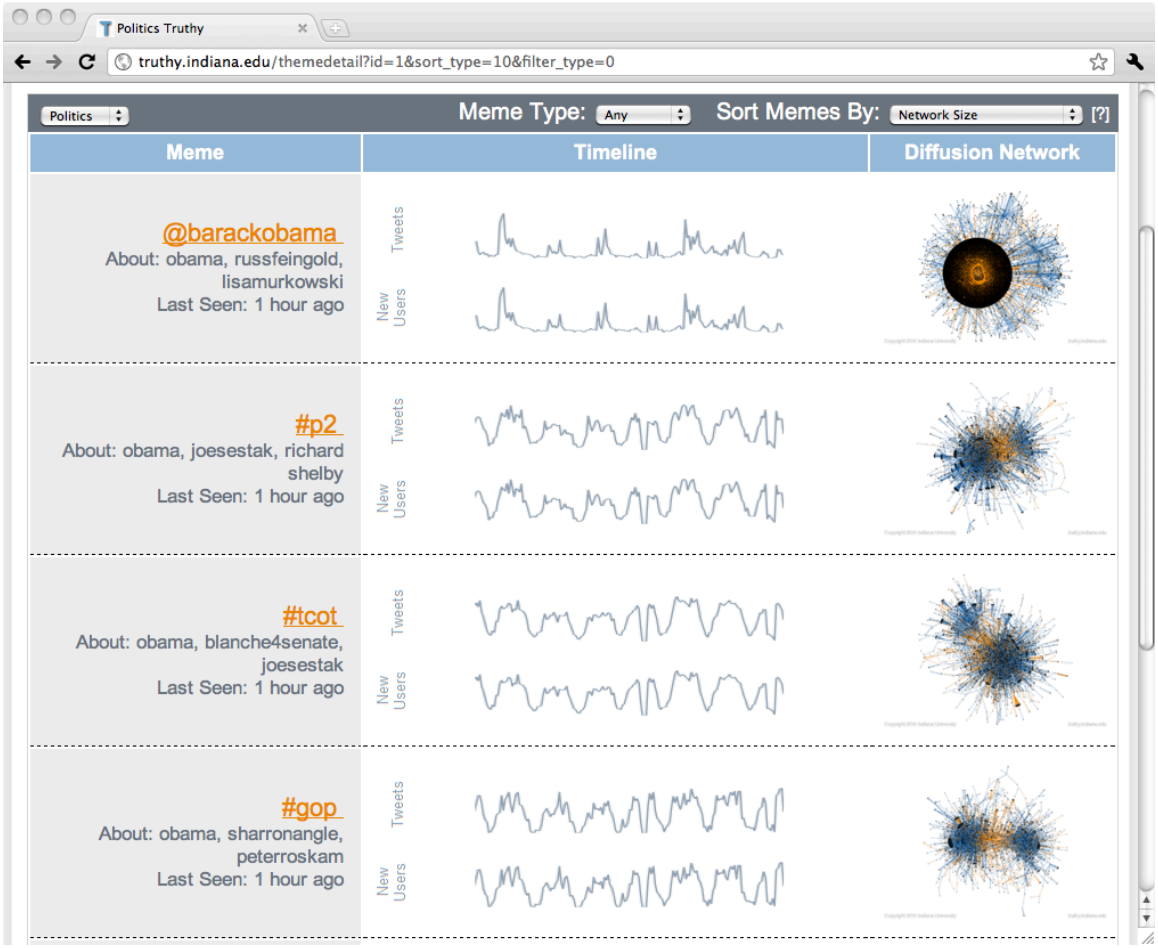


FIGURE 8.5. Screenshot of the Truthy web site meme overview page

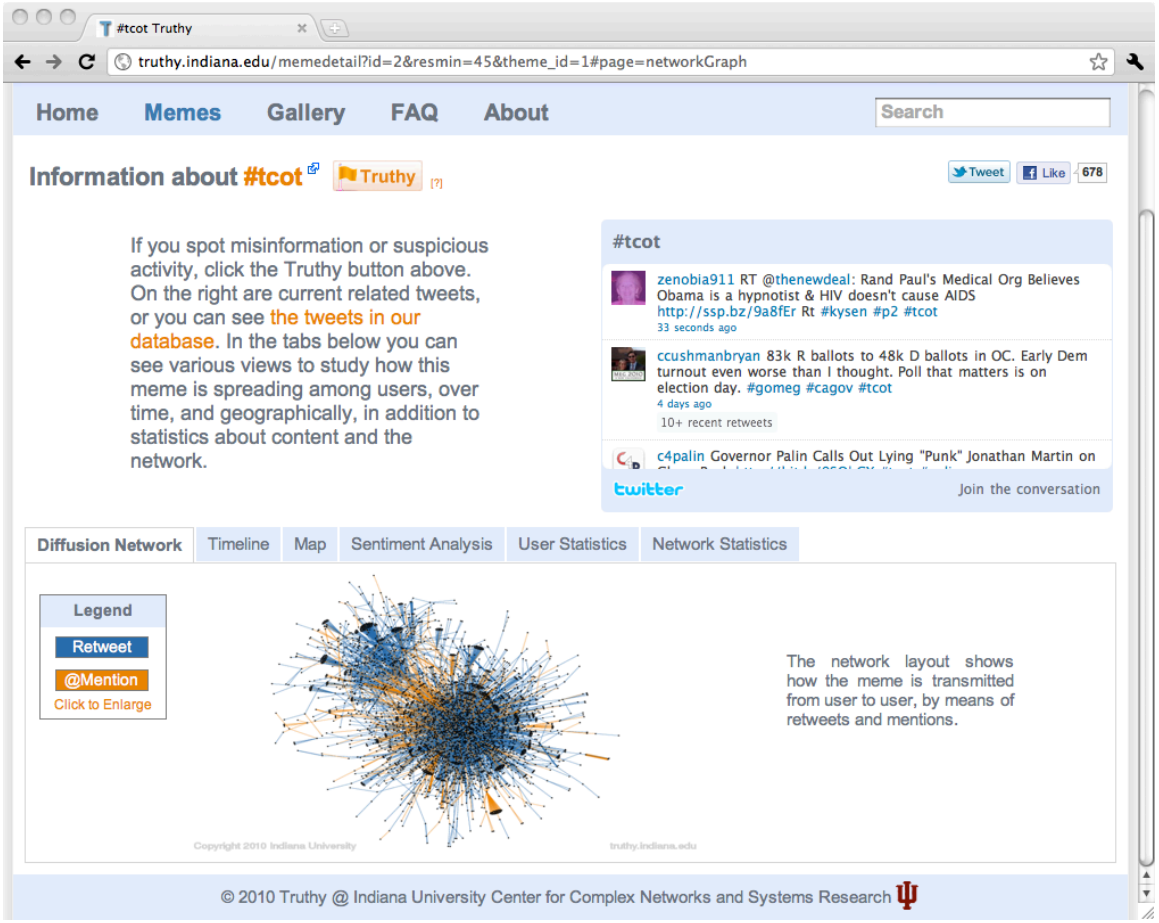


FIGURE 8.6. Screenshots of the Truthy web site meme detail page.

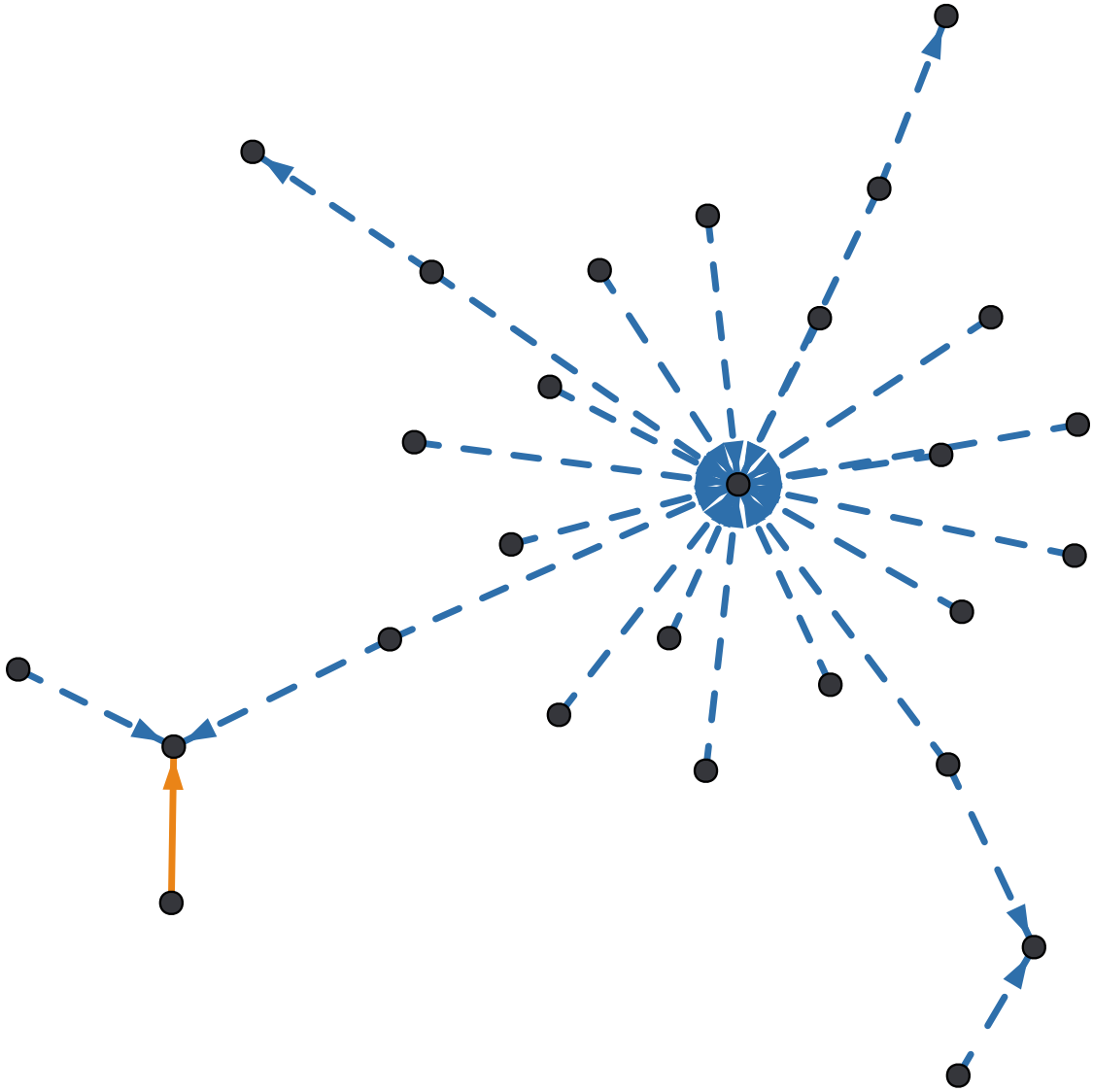
Figure 8.5 and Figure 8.6, respectively. Michael Conover and Snehal Patil did significant work in creating the web site, along with myself.

Users who wish to explore the Truthy database using the Web interface can sort memes according to a variety of ranking criteria, including the size of the largest connected component, number of user annotations, number of users, number of tweets, number of tweets per user, number of retweets, and number of meme injection points — all of which network statistics are pre-computed by the Klatsch framework and stored in the database. This list-based presentation of memes functions as a concise, high-level view of the data, allowing users to examine related keywords, time of most recent activity, tweet volume sparklines and thumbnails of the information diffusion network. At this high level users can examine a large number of memes quickly and subsequently drill down into those that exhibit interesting behavior (Figure 8.5).

Once a user has selected an individual meme for exploration, she is presented with a more detailed presentation of statistical data and interactive visualizations (Figure 8.6). Here the user can examine the statistical data described above, tweets relating the meme of interest, and sentiment analysis data. Additionally users can explore the temporal data through an interactive annotated timeline, inspect a force-directed layout of the meme diffusion network, and view a map of the tweet geo-locations. Upon examining these features, the user is then able to make a decision as to whether this meme is truthy or not, and can indicate her conclusion by clicking a button at the top of the page.

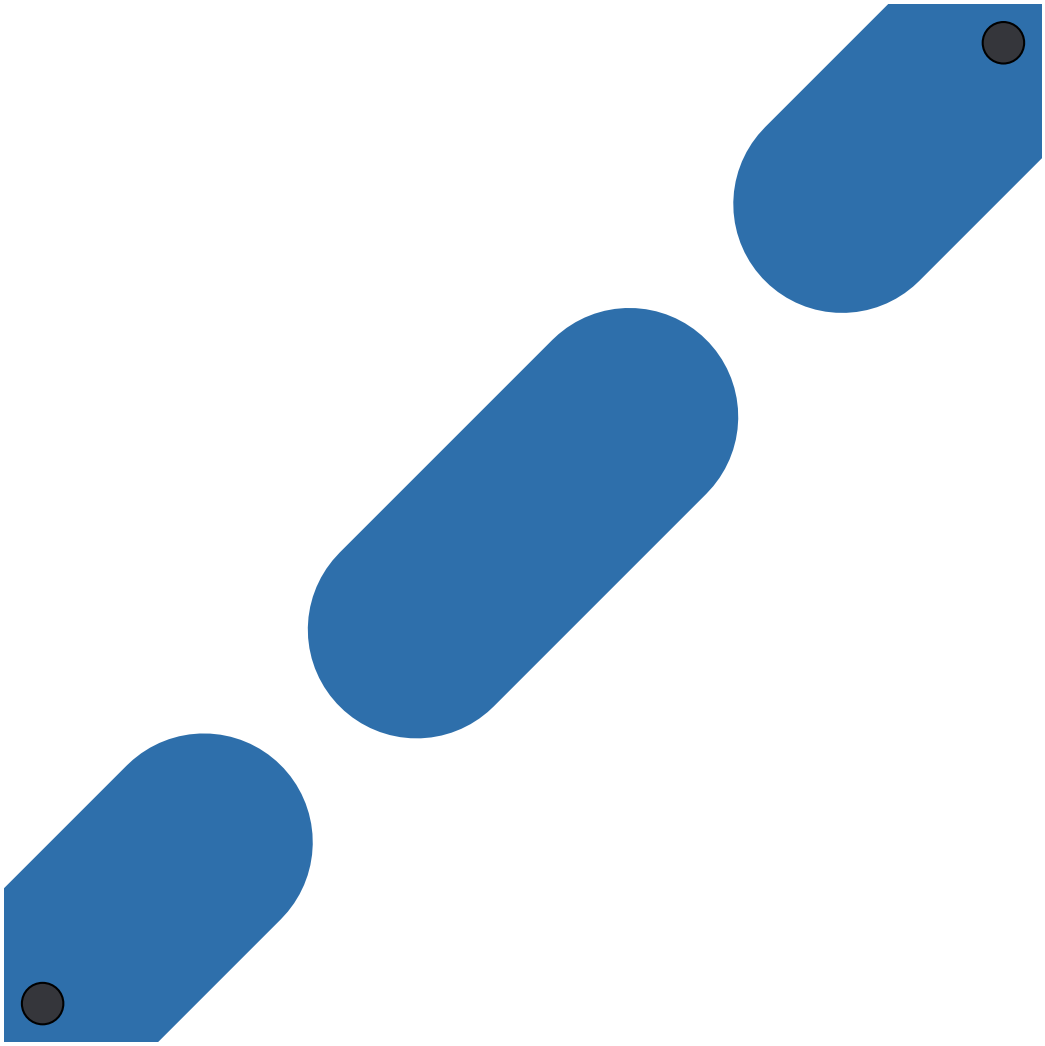
8.4. Examples of Truthy Memes

The Truthy site allowed us to identify several truthy memes. Some of these cases caught the attention of the popular press due to the sensitivity of the topic in the run up to the political elections, and subsequently many of the accounts involved were suspended by Twitter. Below I illustrate a few representative examples.



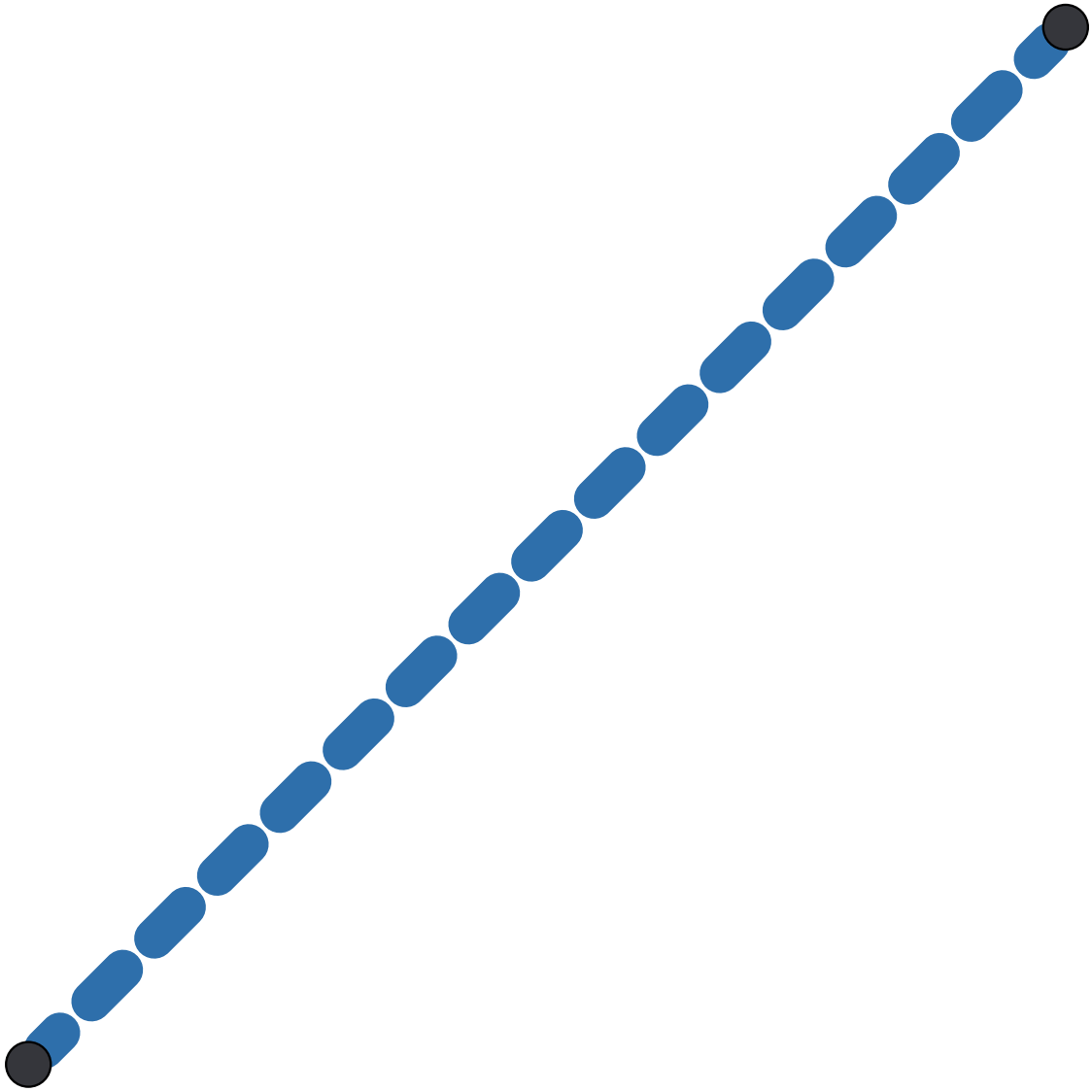
#ampat

The #ampat hashtag is used by many conservative users on Twitter. What makes this meme suspicious is that the bursts of activity are driven by two accounts, @CSteven and @CStevenTucker, which are controlled by the same user, in an apparent effort to give the impression that more people are tweeting about the same topics. This user posts the same tweets using the two accounts and has generated a total of over 41,000 tweets in this fashion.



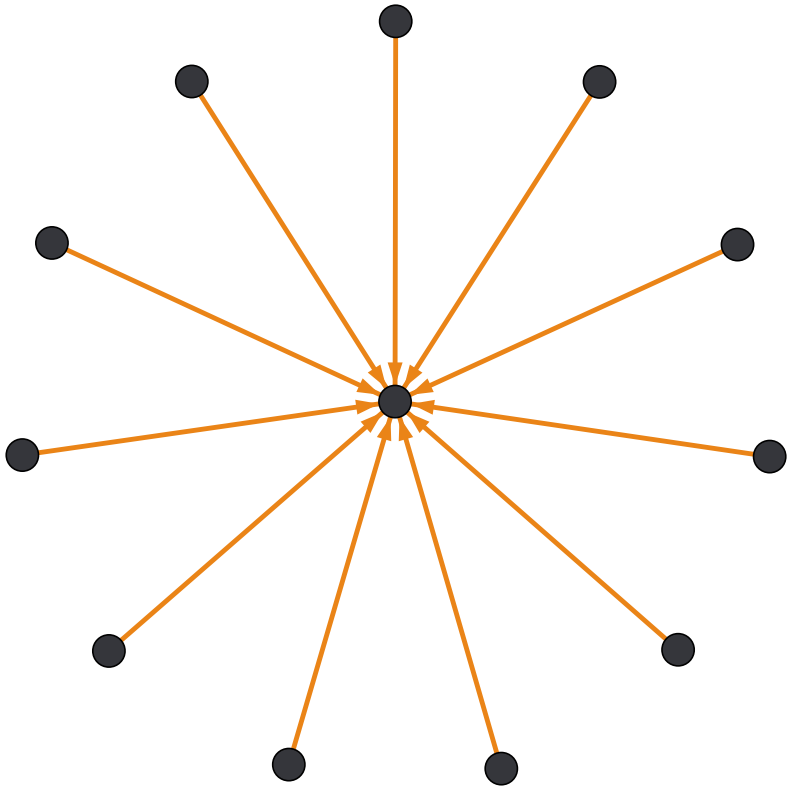
#@PeaceKaren_25

This account did not disclose information about the identity of its owner, and generated a very large number of tweets (over 10,000 in four months). Almost all of these tweets supported several Republican candidates. Another account, `@HopeMarie_25`, had a similar behavior to `@PeaceKaren_25` in retweeting the accounts of the same candidates and boosting the same Web sites. It did not produce any original tweets, and in addition it retweeted all of `@PeaceKaren_25`'s tweets, promoting that account. These accounts had also succeeded at creating a 'twitter bomb': for a time, Google searches for "`gopleader`" returned these tweets in the first page of results. Both accounts were suspended by Twitter by the time of this writing.



gopleader.gov

This meme is the Web site of the Republican Leader John Boehner. It looks truthy because it is boosted by two suspicious accounts described above.



How Chris Coons budget works- uses tax $ 2 attend dinners and fashion shows

This is one of a set of truthy memes smearing Chris Coons, the Democratic candidate for U.S. Senate from Delaware. Looking at the injection points of these memes, we uncovered a network of about ten bot accounts. They inject thousands of tweets with links to posts from the `freedomist.com` Web site. To avoid detection by Twitter and increase visibility to different users, duplicate tweets are disguised by adding different hashtags and appending junk query parameters to the URLs. This works because many URL-shortening services ignore querystrings when processing redirect requests. To generate retweeting cascades, the bots also coordinate mentioning a few popular users. These targets get the appearance of receiving the same news from several different people, and are more likely to think it is true, and spread it to their followers. Most of the bot accounts in this network can be traced back to a single person who runs the `freedomist.com` Web site.

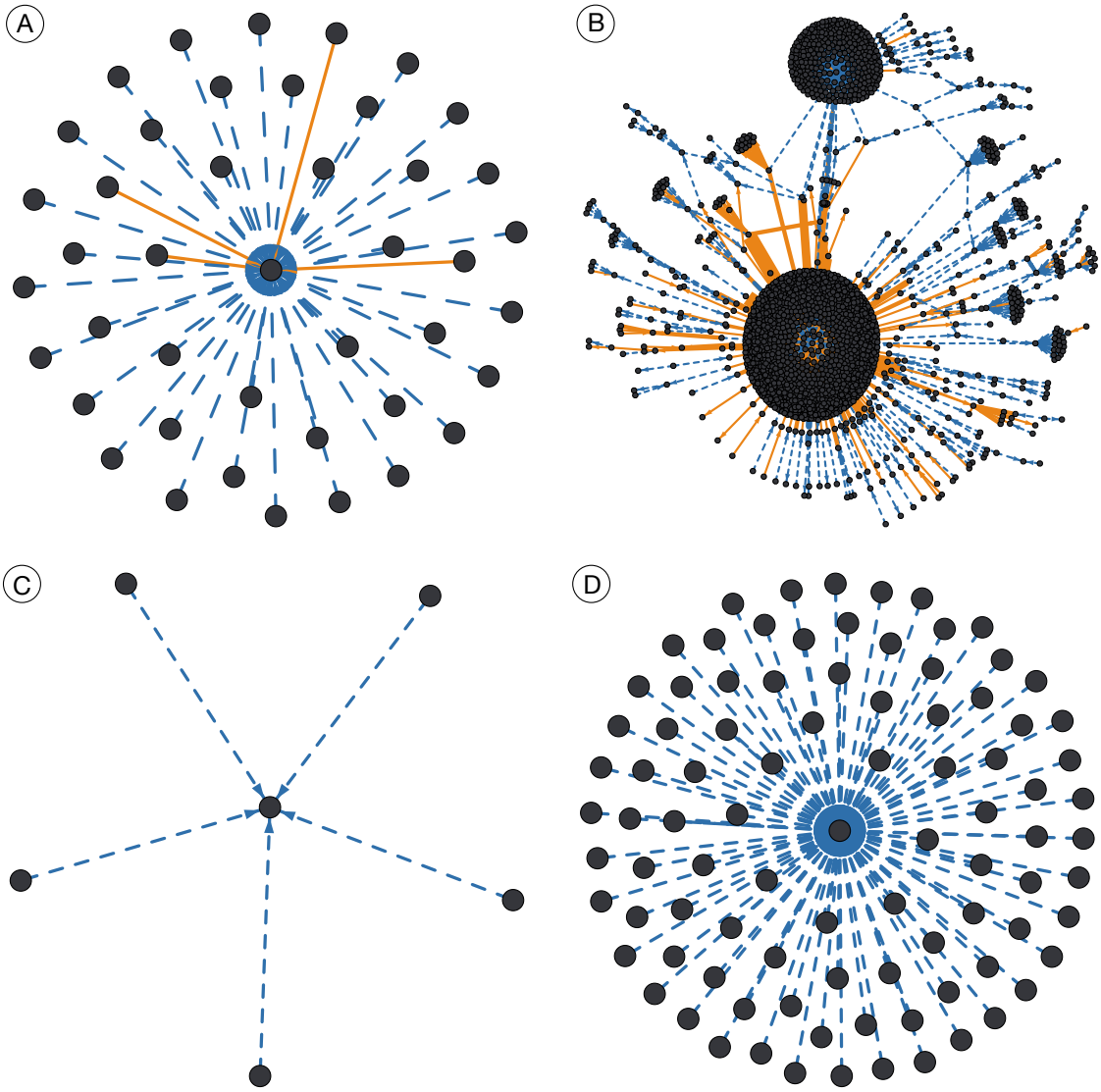


FIGURE 8.7. The diffusion networks of four examples of legitimate memes. Shown are (A) `#truthy`, (B) `@senjohnmccain`, (C) `on.cnn.com/aVMu5y`, (D) "Obama said taxes have gone down during his administration. That's ONE way to get rid of income tax — getting rid of income."

These are just a few instructive examples of characteristically truthy memes our system was able to identify. Two other networks of bots were shut down by Twitter after being detected by Truthy. In one case, we observed the automated accounts using text segments drawn from newswire services to produce multiple legitimate-looking tweets in between the injection of URLs. These instances highlight several of the more general properties of truthy memes detected by our system.

Figure 8.7 also shows the diffusion networks for four legitimate memes. One, `#Truthy`, was injected as an experiment by the NPR Science Friday radio program. Another, `@senjohnmccain`, displays two different communities in which the meme was propagated: one by retweets from `@ladygaga` in the context of discussion on the repeal of the "Don't ask, don't tell" policy on gays in the military, and the other by mentions of `@senjohnmccain`. A gallery with detailed explanations about various truthy and legitimate memes can be found on the Truthy Web site.[1]

8.5. Truthiness Classification

As an application of the analyses performed by the Truthy system, I trained a binary classifier to automatically label legitimate and truthy memes.

I began by enlisting the help of the Truthy team to produce a hand-labeled corpus of training examples in three classes — 'truthy,' 'legitimate,' and 'remove.' We labeled these by presenting viewing memes at random, and placing each meme in one of the three categories. Each reviewer was asked to classify a meme as 'truthy' if a significant portion of the users involved in that meme appeared to be spreading it in misleading ways — e.g., if a number of the accounts tweeting about the meme appeared to be robots or sock puppets, the accounts appeared to follow only other propagators of the meme (clique behavior), or the users engaged in repeated repliest or retweets exclusively with other users who had tweeted the meme. 'Legitimate' memes were described as memes representing normal, legitimate use of Twitter — several non-automated users conversing about a topic. The final category, 'remove,' was to be used for those memes that were in a foreign language, or otherwise did not seem to be related to American politics (`#youth`, for example). These memes were not used in the training or evaluation of classifiers.

[1] `truthy.indiana.edu/gallery`

nodes	Number of nodes
edges	Number of edges
mean_k	Mean degree
mean_s	Mean strength
mean_w	Mean edge weight in largest connected component
max_k(i,o)	Maximum (in,out)-degree
max_k(i,o)_user	User with max. (in,out)-degree
max_s(i,o)	Maximum (in,out)-strength
max_s(i,o)_user	User with max. (in,out)-strength
std_k(i,o)	Std. dev. of (in,out)-degree
std_s(i,o)	Std. dev. of (in,out)-strength
skew_k(i,o)	Skew of (in,out)-degree dist.
skew_s(i,o)	Skew of (in,out)-strength dist.
mean_cc	The mean size of connected components
max_cc	The size of the largest connected component
entry_nodes	Number of unique injections
num_truthy	Number of times 'truthy' button was clicked for the meme
sentiment scores	The six GPOMS sentiment dimensions

TABLE 8.1. Features used in truthy classification.

After we had gathered several hundred annotations we observed an imbalance in our labeled data with less than 10% truthy. Rather than simply resampling, as is common practice in the case of class imbalance, we performed a second round of human annotations on previously-unlabeled memes predicted to be 'truthy' by the classifier trained in the previous round. This bootstrapping process allowed us to manually label a larger portion of truthy memes. The final training dataset

TABLE 8.2. Performance of two classifiers with and without resampling training data to equalize class sizes. All results are averaged based on 10-fold cross-validation.

Classifier	Resampling?	Accuracy	AUC
AdaBoost	No	92.6%	0.91
	Yes	96.4%	0.99
SVM	No	88.3 %	0.77
	Yes	95.6%	0.95

TABLE 8.3. Confusion matrices for a boosted decision stump classifier with and without resampling. The labels on the rows refer to true class assignments; the labels on the columns are those predicted.

	No resampling		With resampling	
	Truthy	Legitimate	Truthy	Legitimate
T	45 (12%)	16 (4%)	165 (45%)	6 (1%)
L	11 (3%)	294 (80%)	7 (2%)	188 (51%)

consisted of 366 training examples — 61 truthy memes and 305 legitimate ones. In those cases where multiple reviewers disagreed on the labeling of a meme, we determined the final label by a group discussion among all reviewers. The dataset is available online.[2]

I used the WEKA machine learning package [**HFH+09**] for classifier training, providing each classification strategy with 31 features about each meme, as shown in Table 8.1. All network features were computed by Klatsch; the six sentiment scores were computed by the GPOMS sentiment analysis method of Bollen *et al.* [**BMP10**]. I experimented with two classifiers: AdaBoost with DecisionStump, and SVM. As the number of instances of truthy memes was still less than instances of legitimate ones, I also experimented with resampling the training data to balance the classes prior to classification. The performance of the classifiers is shown in Table 8.2, as evaluated by their accuracy and the area under their ROC curves. In all cases these preliminary results are quite encouraging, with accuracy around or above 90%. The best results are obtained by AdaBoost with

[2] `cnets.indiana.edu/groups/nan/truthy`

χ^2	Rank	Feature
230 ± 4	1.0 ± 0.0	`mean_w`
204 ± 6	2.0 ± 0.0	`mean_s`
188 ± 4	4.3 ± 1.9	`edges`
185 ± 4	4.4 ± 1.1	`skew_ko`
183 ± 5	5.1 ± 1.3	`std_si`
184 ± 4	5.1 ± 0.9	`skew_so`
180 ± 4	6.7 ± 1.3	`skew_si`
177 ± 4	8.1 ± 1.0	`max_cc`
174 ± 4	9.6 ± 0.9	`skew_ki`
168 ± 5	11.5 ± 0.9	`std_ko`

TABLE 8.4. Top 10 most discriminative features, according to a χ^2 analysis under 10-fold cross validation. Intervals represent the variation of the χ^2 or rank across the folds.

resampling. Table 8.3 further shows the confusion matrices for AdaBoost. In this task, false negatives (truthy memes incorrectly classified as legitimate) are less desirable than false positives. In the worst case, the false negative rate is less than 5%. Table 8.4 shows the 10 most discriminative features, as determined by χ^2 analysis. Network features appear to be more discriminative than sentiment scores or the few user annotations that we collected.

8.6. Discussion

This chapter discussed Truthy, a system for the visualization of the spread of political memes on Twitter. Truthy's goal is to be useful for helping detect astroturfing campaigns in the context of U.S. political elections. The network features computed by Klatsch and made available through Truthy show promise for accurately detecting truthy memes. Using this system we have been able to identify a number of genuinely truthy memes. Though few of these exhibit the explosive growth characteristic of true viral memes, they are nonetheless clear examples of coordinated attempts to deceive Twitter users. Truthy memes are often spread initially by bots, causing them to exhibit pathological diffusion graphs relative to what is observed in the case of organic memes. These

graphs can take many forms, including high numbers of unique injection points with few or no connected components, strong star-like topologies characterized by high average degree, and most tellingly large edge weights between dyads in graphs that exhibit either of the above properties.

A major component of Truthy is its system to tracking and filtering memes in real time. I describe the implementation of this subsystem, in a way that is efficient and accurate. It runs in real time, and retains many useful memes while keeping the total size of the meme set manageable.

The accuracy scores I observe in the classification task are quite high. I hypothesize that this performance is partially explained by the fact that a consistent proportion of the memes were failed attempts of starting a cascade. In these cases the networks reduced to isolated injection points or small components, resulting in trivial network features that allowed for easy classification.

Despite the fact that many of the memes discussed in this paper are characterized by small diffusion networks, it is important to note that this is the stage at which such attempts at deception must be identified. Once one of these attempts is successful at gaining the attention of the community, it will quickly become indistinguishable from an organic meme. Therefore, the early identification and termination of accounts associated with astroturf memes is critical.

The Truthy team intends to add more views to the website, including views on the users, such as the ages of the accounts, and tag clouds to interpret the sentiment analysis scores. We need to collect more labeled data about truthy memes in order to achieve more meaningful classification results, and will also explore the use of additional features in the classifiers, such as account ages for the most active users in a meme, and reputation features for users based on the memes to which they contribute. Another important area to address is that of sampling bias, since the properties of the sample made available in the Twitter gardenhose are currently unknown. To explore this, we intend to track injected memes of various sizes and with different topological properties of their diffusion graphs.

CHAPTER 9

CONCLUSION

9.1. Summary and Discussion

In this dissertation I have described several results related to the popularity of ideas, and the behaviors of the people who create and spread them. Chapter 5 presented a number of exploratory results on the behavior of users browsing Wikipedia. I find that linked pages are correlated in the traffic they receive, and that much of this correlation is due to people navigating from one to the other along a link between the two. This is consistent with the view that browsing from one page to another is a common use case for Wikipedia (in contrast with, for instance, navigating to a particular Wikipedia page from a web search). Focusing on dramatic shifts in the popularity of pages, I find that these are mainly driven by external sources. Chapter 6 explores these bursty popularity dynamics in more detail, measuring their size and time distribution. This analysis paints a picture of a system in which dramatic shifts of popularity are possible, with no characteristic time frequency for when these shifts might arrive. Here also I describe the inadequacy of rich-get-richer models in capturing these bursty dynamics, and describe a model that does (the *rank-shift* model).

In Chapters 7 and 8 I look at the spread of ideas through the users who propagate them. Here I can examine not just how many people have learned about an idea, but also *who* those people are and how the idea spread from person to person. In Chapter 7 I examine the communities formed

by people discussing American politics, and find that people on each end of the political spectrum tend to associate with each other preferentially.

Finally I devote Chapter 8 to a case study on a system designed to visualize the spread of memes on Twitter, and help identify memes which might represent misinformation campaigns. This identification can be performed solely on machine learning features derived from the *structural information* of the spread of the meme between users, with accuracy greater than 90%.

9.2. Future work

I would divide the future work suggested by the results I present into several broad categories: the *modeling* and *prediction* of the dynamical behavior of online popularity, and a category encompassing exploratory analysis of the social network results I present in Chapters 7 and 8.

9.2.1. Modeling and predicting online popularity

The model developed in response to the bursty popularity dynamics in Chapter 6, while able to capture these dynamics, is very simple [**RFF$^+$10**]. While this simplicity is by design, there are other observed features of the data that the model cannot capture. For instance, I show in Chapter 5 that many kinds of correlations in traffic exist between pages, based on their link relationship and content similarity. More complex modeling strategies could include an agent-based model where agents are modeled by a level of *activity*, in page views per unit time, and an *interest vector* defining the kinds of pages they would like to see. Pages could be modeled by their link structure and a *topic vector* (from the same space as the agents' interest vectors).

Taking modeling a step farther, there remains the problem of *predicting* the popularity of pages in advance. While I show in Chapter 5 that many surges in popularity are due to news events which may be impossible to predict, this leaves open the possibility of predicting popularity bursts which arise due to network effects, or predicting the popularity of non-bursty pages. I have performed some initial experiments with using Fourier series to model and predict the popularity of pages whose popularity patterns looked cyclical; however, these require further work. Even these pages, seemingly the easiest to predict, show complex popularity dynamics.

9.2.2. Political speech online

The preliminary results in the field of online political speech in Chapters 7 and 8 suggest several avenues for future work. One obvious such avenue is the expansion of these results into subject areas other than American politics. I have performed some preliminary experiments which suggest that diffusion networks for political memes are structurally different than for random memes; specifically, they naturally have a smaller number of clusters (as found by a clustering algorithm which also attempts to determine the 'best' number of clusters). It is possible that such an observation could be used to identify political memes without considering content at all.

It would be interesting to model the growth of the diffusion network for a particular meme, which is related to modeling the retweet probabilities of individual users. Similar to the Wikipedia or Web browsing model mentioned above, a user might have an *interest vector*, which essentially encodes their probabilities of retweeting memes in various topics. Each user might also have an *influence factor*, which might encode the fact that some users are more respected sources of information than others. Memes would then be introduced to a network of interconnected users, and their diffusion could be tracked. The problem of computed a user's influence in a social network has been studied [**HDD11, RGAH11**]. There has also been limited progress on the related problem of predicting message spread [**GAC$^+$10**]. The spread of memes across a network might also be related to the growth of the network — it is intuitive that someone might choose to follow another person who they notice is often an originator of interesting material. Capturing this relationship would require a new class of models that track both the diffusion of memes across existing social links, as well as the creation of new diffusion conduits.

BIBLIOGRAPHY

[AA05] Eytan Adar and Lada A. Adamic. Tracking information epidemics in blogspace. *2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 0:207–214, 2005.

[AFL+10] Sean Aday, Henry Farrel, Marc Lynch, John Sides, John Kelly, and Ethan Zuckerman. Blogs and bullets: New media in contentious politics. Technical report, United States Institute of Peace, 2010.

[AG05] Lada Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proc. 3rd Intl. Workshop on Link Discovery (LinkKDD)*, pages 36–43, 2005.

[AH10] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. Technical Report arXiv:1003.5699, CoRR, 2010.

[AJB99] R Albert, H Jeong, and A-L Barabási. Diameter of the World Wide Web. *Nature*, 401(6749):130–131, 1999.

[AMC07] R. B. Almeida, B. Mozafari, and Junghoo Cho. On the evolution of Wikipedia. In *ICWSM '07: Proceedings of the International Conference on Weblogs and Social Media*, March 2007.

[BA99] Albert-Laszlo Barabási and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[Bar05] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207—211, 2005.

[BB01] Ginestra Bianconi and Albert-László Barabási. Bose-Einstein condensation in complex networks. *Phys. Rev. Lett.*, 86(24):5632–5635, Jun 2001.

[BBV04] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. Traffic-driven model of the world wide web graph. In Stefano Leonardi, editor, *Algorithms and Models for the Web-Graph*, volume 3243 of *Lecture Notes in Computer Science*, pages 56–67. Springer Berlin / Heidelberg, 2004.

[BC00] BE Brewington and G Cybenko. Keeping up with the changing Web. *IEEE Computer*, 33(5):52–58, 2000.

[BCD+06] Luciana S. Buriol, Carlos Castillo, Debora Donato, Stefano Leonardi, and Stefano Millozzi. Temporal analysis of the Wikigraph. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 45–51, Washington, DC, USA, 2006. IEEE Computer Society.

[Ben06] Yochai Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.

[BKM+00] A Broder, SR Kumar, F Maghoul, P Raghavan, S Rajagopalan, R Stata, A Tomkins, and J Wiener. Graph structure in the Web. *Computer Networks*, 33(1–6):309–320, 2000.

[BMP10] Johan Bollen, Huina Mao, and Alberto Pepe. Determining the public mood state by analysis of microblogging posts. In *Proc. of the Alife XII Conf.* MIT Press, 2010.

[BS03] Marian Boguna and Romualdo P. Satorras. Class of correlated random networks with hidden variables. *Phys. Rev. E*, 68:036112, 2003.

[BYP06] Ricardo Baeza-Yates and Barbara Poblete. Dynamics of the Chilean web structure. *Comput. Netw.*, 50(10):1464–1473, 2006.

[Cal99] Pável Pereira Calada. The WBR-99 collection. Technical report, Departamento de Computação, Universidade Federal de Minas Gerias, 1999.

[CBBV07] Vittoria Colizza, Marc Barthélemy, Alain Barrat, and Alessandro Vespignani. Epidemic modeling in complex realities. *Comptes Rendus Biologies*, 330(4):364 – 374, 2007.

[CRF+11] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Political polarization on twitter. In *International Conference on Weblogs and Social Media*. International Conference on Weblogs and Social Media, July 2011.

[CS08] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proc. Natl. Acad. Sci. USA*, 105(41):15649–15653, 2008.

[CSC+06] A. Capocci, V. D. P. Servedio, F. Colaiori, L. S. Buriol, D. Donato, S. Leonardi, and G. Caldarelli. Preferential attachment in the growth of social networks: The internet encyclopedia Wikipedia. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3):036116, 2006.

[CSN07] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. Technical report, arXiv:0706.1062v1 [physics.data-an], 2007.

[DAL+06] Z. Dezso, E. Almaas, A. Lukacs, B. Racz, I. Szakadat, and A.L. Barabasi. Dynamics of information access on the Web. *Phys. Rev. E*, 73:066132, 2006.

[DMS00] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Phys. Rev. Lett.*, 85(21):4633–4636, Nov 2000.

[dSP76] DJ de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *J. Amer. Soc. Inform. Sci.*, 27:292–306, 1976.

[Ear10] P. Earle. Earthquake Twitter. *Nature Geoscience*, 3:221, 2010.

[ER60] P. Erdős and A Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[FD08] Henry Farrell and Daniel Drezner. The power and politics of blogs. *Public Choice*, 134(1):15–30, January 2008.

[FFM06] Santo Fortunato, Alessandro Flammini, and Filippo Menczer. Scale-free network growth by ranking. *Physical Review Letters*, 96(21):218701, 2006.

[FFMV06] S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. Topical interests and the mitigation of search engine bias. *Proc. Natl. Acad. Sci. USA*, 103(34):12684–12689, 2006.

[FKP02] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet (extended abstract). In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP*, pages 110–122, 2002.

[FMNW03] D Fetterly, M Manasse, M Najork, and JL Wiener. A large-scale study of the evolution of Web pages. In *Proc. 12th International World Wide Web Conference*, 2003.

[FR91] Thomas M J Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 21(11):1129–1164, 1991.

[GAC+10] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. Outtweeting the Twitterers — Predicting Information Cascades in Microblogs. In *3rd Workshop on Online Social Networks (WOSN'10)*, 2010.

[GPV11] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter: validation of dunbar's number. In preparation, 2011.

[GR44] B. Gutenberg and C.F. Richter. Frequency of earthquakes in California. *Bull. Seismol. Soc. Am.*, 34:185–188, 1944.

[GS66] D. M Green and J. A Swets. *Signal detection theory and psychophysics*. John Wiley and Sons Inc, 1966.

[HA85] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. 10.1007/BF01908075.

[HDD11] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 57–58, New York, NY, USA, 2011. ACM.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations*, 11(1):10–18, 2009.

[KKR+99] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[Kle02] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2002.

[KNRT03] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *Proc. 12th International World Wide Web Conference*, pages 568–576, 2003.

[LAH06] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237, New York, NY, USA, 2006. ACM.

[LBKT08] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470, New York, NY, USA, 2008. ACM.

[LdSAH07] Pedro G. Lind, Luciano R. da Silva, José S. Andrade, and Hans J. Herrmann. Spreading gossip in social networks. *Phys. Rev. E*, 76(3):036117, Sep 2007.

[LF06] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 631–636, 2006.

[LKF07] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007.

[Man97] B. B. Mandelbrot. *Fractals and Scaling in Finance: Discontinuity, Concentration, Risk*, volume E of *Selecta*. Springer, 1997.

[Men04] Filippo Menczer. Evolution of document networks. *Proceedings of the National Academy of Sciences*, 101(1):5261–5265, 2004.

[MKM10] Michael Mathioudakis, Nick Koudas, and Peter Marbach. Early online identification of attention gathering items in social media. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 301–310, New York, NY, USA, 2010. ACM.

[MM10] E. Mustafaraj and P. Metaxas. From obscurity to prominence in minutes: Political speech and real-time search. In *Proc. Web Science: Extending the Frontiers of Society On-Line (WebSci)*, page 317, 2010.

[MMF+08] Mark R. Meiss, Filippo Menczer, Santo Fortunato, Alessandro Flammini, and Alessandro Vespignani. Ranking web sites with real user traffic. In *WSDM '08: Proceedings of the international conference on Web search and Web data mining*, pages 65–76, New York, NY, USA, 2008. ACM.

[MMV05] Mark Meiss, Filippo Menczer, and Alessandro Vespignani. On the lack of typical behavior in the global web traffic network. In *WWW*, pages 510–518, 2005.

[MMV11] Mark Meiss, Filippo Menczer, and Alessandro Vespignani. Properties and evolution of internet traffic networks from anonymized flow data. *ACM Trans. Internet Technol.*, 10:15:1–15:23, March 2011.

[Mor00] Stephen Morris. Contagion. *The Review of Economic Studies*, 67(1):pp. 57–78, 2000.

[NCO04] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the Web? The evolution of the Web from a search engine perspective. In *Proc. 13th Intl. Conf. on World Wide Web*, pages 1–12. ACM Press, 2004.

[New06a] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(3):036104, Sep 2006.

[New06b] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Omo94] F. Omori. On the after-shocks of earthquakes. *J. Coll. Sci. Imp. Univ. Japan*, 7:111–200, 1894.

[PSV01] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86(14):3200–3203, Apr 2001.

[RAK07] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76(3):036106, Sep 2007.

[Ran71] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):pp. 846–850, 1971.

[RCM+10] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. Detecting and tracking the spread of astroturf memes in microblog streams. Technical Report arXiv:1011.3768 [cs.SI], CoRR, 2010.

[RFF+10] Jacob Ratkiewicz, Santo Fortunato, Alessandro Flammini, Filippo Menczer, and Alessandro Vespignani. Characterizing and modeling the dynamics of online popularity. *Phys. Rev. Lett.*, 105(15):158701, Oct 2010.

[RGAH11] Daniel M. Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 113–114, New York, NY, USA, 2011. ACM.

[RMF+10] Jacob Ratkiewicz, Filippo Menczer, Santo Fortunato, Alessandro Flammini, and Alessandro Vespignani. Traffic in Social Media II: Modeling Bursty Popularity. In *Proc. of the International Symposium on Social Intelligence and Networking (SIN-10)*. IEEE, 2010.

[SAB+96] Michael H. R. Stanley, Luis A. N. Amaral, Sergey V. Buldyrev, Shlomo Havlin, Heiko Leschhorn, Philipp Maass, Michael A. Salinger, and H. Eugene Stanley. Scaling behaviour in the growth of companies. *Nature*, 379(6568):804–806, 02 1996.

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[SDW06] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science*, 311(5762):854–856, 2006.

[SH08] Gabor Szabo and Bernardo A. Huberman. Predicting the popularity of online content. Technical report, arXiv:0811.0405v1 [cs.CY], 2008.

[Sim55] Herbert A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425—440, 1955.

[SMB+07] M. Angeles Serrano, Ana Maguitman, Marian Boguna, Santo Fortunato, and Alessandro Vespignani. Decoding the structure of the WWW: A comparative analysis of Web crawls. *ACM Trans. Web*, 1(2):10, 2007.

[SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW '10: Proc. of the 19th international Conf. on World wide web*, pages 851–860, New York, NY, USA, 2010. ACM.

[SP06] Sitabhra Sinha and Raj Kumar Pan. *How a "Hit" is Born: The Emergence of Popularity from the Dynamics of Collective Choice*. Wiley-VCH Verlag GmbH & Co. KGaA, 2006.

[Sun07] Cass R. Sunstein. *Republic.com 2.0*. Princeton University Press, August 2007.

[TW06] Don Tapscott and Anthony D. Williams. *Wikinomics: How Mass Collaboration Changes Everything*. Portfolio Hardcover, December 2006.

[vR79] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979.

[VZ98] A. Vespignani and S. Zapperi. How self-organized criticality works: A unified mean field picture. *Phys. Rev. E*, 57:6345–6362, 1998.

[Wat99] Duncan J. Watts. Networks, dyanamics, and the small-world phenomenon. *American Journal of Sociology*, 105(2):493–527, 1999.

[WH07] Fang Wu and Bernardo A. Huberman. Novelty and collective attention. *Proc. Natl. Acad. Sci. USA*, 104(45):17599–17601, 2007.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 06 1998.

[Zac77] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):pp. 452–473, 1977.

[ZBSD06] V. Zlatic, M. Bozicevic, H. Stefancic, and M. Domazet. Wikipedias: Collaborative web-based encyclopedias as complex networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(1):016115, 2006.

[Zip49] George K. Zipf. *Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology*. Addison-Wesley, 1949.

Jacob Ratkiewicz

506 E. Allen St.
Bloomington, IN 47401 jpr@cs.indiana.edu
(812) 369-0980 http://www.cs.indiana.edu/~jpr

Biographical Sketch

Jacob Ratkiewicz is a PhD candidate in the Indiana University School of Informatics & Computing, working with Professor Filippo Menczer. His current research interests include the topology of the Web, and mining traffic data from the Web to model human Web behavior and forecast upcoming trends.

Education

Ph. D. (ABD)	Computer Science	Indiana University Bloomington	expected 2011
M.S.	Computer Science	Indiana University Bloomington	2005
B.S.	Computer Science	Indiana University at South Bend	2003 (highest distinction)

Work Experience

Google, Inc. Software engineering intern in Research.	Summer 2010
Research Assistant, (Web Topology project, PI Menczer). Leveraging several large sources of data about Web traffic, in particular traffic to Wikipedia articles, in order to model popularity and trends in the Web. A major focus of research is developing tools to predict future popularity, or detect trending topics early.	Fall 2009 — Present
Research Assistant (GiveALink project, PI Menczer). Using machine learning methods to manage tag spam in the GiveALink social bookmarking site. Also responsible for some general site maintenance and programming duties.	Summer 2009
Research Staff (Complex Networks Lagrange Lab, ISI Foundation). Pursuing ongoing research under Filippo Menczer and Alessandro Vespignani. Projects included Domrank, a text retrieval method based on the technique of decomposing a web page into its Document-Object Model and then using graph-theoretic ranking metrics (such as PageRank) to rank the resulting nodes, and measuring changes in web topology over time using data from Wikipedia and a large crawl of the Web over several years.	2007 — 2008
Google, Inc. Software engineering intern in Internal Applications team.	Summer 2007
RavenWhite, Inc. Consultant. Responsibilities including software and technology development.	2006 — 2007

Other Research Projects

Click Fraud: With Markus Jakobsson and other graduate students, researching novel methods for click fraud, and countermeasures against them. Experimentally measuring success rates of various click fraud strategies.

Large Graph Library: With Filippo Menczer, translating the Webgraph large graph library (`http://http://webgraph.dsi.unimi.it/`) from Java to C++ in order to extend its capabilities to larger graphs.

Teaching Experience

Introduction to Programming I. Course instructor. Responsible for all aspects of the course (curriculum, textbook, assignments, tests, grading).

Introduction to Programming II. Associate instructor. Responsible for office hours and grading.

Information Infrastructure II. Associate instructor. Responsible for office hours, weekly labs, and grading.

Introduction to Algorithms Design and Analysis. Associate instructor. Responsible for office hours and grading.

Search Informatics. Associate instructor. Responsible for office hours and grading.

Algorithms Design and Analysis. Associate instructor. Responsible for office hours and grading.

Specification and Verification. Associate instructor. Responsible for office hours, grading, and overseeing student presentations.

Advanced Operating Systems. Associate instructor. Responsible for overseeing a large student project, a weekly discussion section, grading, and some lectures.

Scientific Computing. Associate instructor. Responsible for office hours, grading, and some lectures.

Web Mining. Associate instructor. Responsible for office hours, grading, and overseeing a large student project.

Publications

Refereed Proceedings

- M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, A. Flammini, F. Menczer. Political Polarization on Twitter. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*, July 2011 (to appear).
- J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, F. Menczer. Detecting and Tracking Political Abuse in Social Media. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*, July 2011 (to appear).
- J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)*. ACM, New York, NY, USA, 249-252.
- J. Ratkiewicz, F. Menczer, S. Fortunato, A. Flammini, A. Vespignani. Characterizing and modeling the dynamics of online popularity. *Phys. Rev. Let 105, 158701*
- J. Ratkiewicz, A. Flammini, F. Menczer. Traffic in Social Media I: Paths through information networks. In *Proceedings of the Second International IEEE Conference on Social Computing (SocialCom)*, pp.452-458, 20-22 Aug. 2010
- J. Ratkiewicz, F. Menczer, S. Fortunato, A. Flammini, A. Vespignani. Traffic in Social Media II: Modeling Bursty Popularity. In *Proceedings of the Second International IEEE Conference on Social Computing (SocialCom)*, pp.393-400, 20-22 Aug. 2010
- J. Ratkiewicz, F. Menczer. Text snippets from the domgraph. In *Proceedings of the SIGIR 2008 Workshop on Focused Retrieval, pages 45 – 50, 2008*
- M. Jakobsson, M. Gandhi, and J. Ratkiewicz, Badvertisements: Stealthy Click Fraud with Unwitting Accessories. APWG 2006 eCrime Researcher's Summit.
- M. Jakobsson, J. Ratkiewicz, Designing Ethical Phishing Experiments: A Study of (ROT13) rOnl Query Features. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*. ACM, New York, NY, USA, 513-522.
- D. Vrajitoru, J. Ratkiewicz, Evolutionary Sentence Combination for Chatterbots. The IASTED International Conference on Artificial Intelligence and Applications (AIA 2004).

Book Chapters

- J. Ratkiewicz, "Content Injection Tutorial." in "Phishing and Countermeasures", M. Jakobsson, S. Myers (eds.), Wiley, 2006.
- J. Ratkiewicz, "Attacking eBay Users with Queries." in "Phishing and Countermeasures", M. Jakobsson, S. Myers (eds.), Wiley, 2006.

Awards and Honors

- Best Speaker, Communications Fraud Control Alliance 21st Annual Meeting and Conference, St. Louis, MO (June 27-29, 2006).
- ACM Intercollegiate Programming Contest 2003 — Team placed 20/113 nationally, 2nd regionally (IUSB Titaniums).
- Excellence Award in Computer Science — IU South Bend
- Graduation with Highest Distinction — IU South Bend (B.S.)

Program Committees

- ACM Conference on Hypertext and Hypermedia 2008, 2009, 2010

Professional Activities

- Computer Science Department Graduate Student Association President — 2005
- Computer Science Department Graduate Student Association Social Chair — 2004
- Computer Science Department Graduate Student Association Secretary — 2003
- ACM Student Chapter President — 2001, 2002

Citizenship

United States of America

References

- Filippo Menczer, Associate Professor of Informatics and Computer Science, and Associate Director of the Center for Complex Networks and Systems Research, Indiana University. `fil@indiana.edu`
- Alessandro Vespignani, Professor of Informatics and Computing, and Director of the Center for Complex Networks and Systems Research at Indiana University; also Research Director, Institute for Scientific Interchange Foundation, Turin, Italy. `alexv@indiana.edu`
- Markus Jakobsson, Principal Scientist, PARC; also Adjunct Professor of Informatics at Indiana University. `markus.jakobsson@parc.com`

Further references available upon request.