The Political Blogosphere and the 2004 U.S. Election: Divided They Blog

Lada A. Adamic
HP Labs
1501 Page Mill Road Palo Alto, CA 94304
lada.adamic@hp.com

Natalie Glance
Intelliseek Applied Research Center
5001 Baum Blvd. Pittsburgh, PA 15217
nglance@intelliseek.com

ABSTRACT

In this paper, we study the linking patterns and discussion topics of political bloggers. Our aim is to measure the degree of interaction between liberal and conservative blogs, and to uncover any differences in the structure of the two communities. Specifically, we analyze the posts of 40 "A-list" blogs over the period of two months preceding the U.S. Presidential Election of 2004, to study how often they referred to one another and to quantify the overlap in the topics they discussed, both within the liberal and conservative communities, and also across communities. We also study a single day snapshot of over 1,000 political blogs. This snapshot captures blogrolls (the list of links to other blogs frequently found in sidebars), and presents a more static picture of a broader blogosphere. Most significantly, we find differences in the behavior of liberal and conservative blogs, with conservative blogs linking to each other more frequently and in a denser pattern.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: [Data Mining]; J.4 [**Social and Behavioral Sciences**]: [Sociology]; G.2.2 [**Graph Theory**]: [Network Problems]

Keywords

political blogs, social networks, link analysis

1. INTRODUCTION

The 2004 U.S. Presidential Election was the first presidential election in the United States in which blogging played an important role. Although the term weblog was coined in 1997, it was not until after 9/11 that blogs gained readership and influence in the U.S. According to a report from the Pew Internet & American Life Project[1], by January 2005 one in four internet users in the U.S. read weblogs, but 62% of them still did not know what a weblog was. During the presidential election campaign many Americans turned to the Internet to stay informed about politics, with 9% of Internet users saying that they read political blogs "frequently" or "sometimes"[2]. Indeed, political blogs showed a large growth in readership in the months preceding the election.[3]

Recognizing the importance of blogs, several candidates and political parties set up weblogs during the 2004 U.S. Presidential campaign. Notably, Howard Dean's campaign was particularly successful in harnessing grassroots support using a weblog as a primary mode for publishing dispatches from the candidate to his followers. The Democratic and Republican parties further signaled the established position of blogs in political discourse by credentialing a number of bloggers to cover their nominating conventions as journalists. This lead to the creation of sites to aggregate the content of these blogs during the national conventions: `conventionbloggers.com` for the RNC, `cyberjournalist.net` for the DNC[4]. Other sites adapted to keep track of an ever proliferating mass of political blogs by creating specialized political blog search engines, aggregate feeds and search analytics, including Feedster (`politics.feedster.com`), BlogPulse (`politics.blogpulse.com`) and Technorati (`politics.technorati.com`).

Weblogs may be read by only a minority of Americans, but their influence extends beyond their readership through their interaction with national mainstream media. During the months preceding the election, there were several cases in which political blogs served to complement mainstream media by either breaking stories first or by fact-checking news stories. For example, bloggers first linked to Swiftvets.com's anti-Kerry video in late July and kept the accusations alive, until late August, when John Kerry responded to their claims, bringing mainstream media coverage.[5] In another example, bloggers questioned CBS News' credibility over the memos purportedly alleging preferential treatment toward President Bush during the Vietnam War. Powerline broke the story on September 9th[6], launching a flurry of discussions across political blogs and beyond. Dan Rather apologized later in the month.

[1] `http://www.pewinternet.org/PPF/r/144/report_display.asp`

[2] `http://www.pewinternet.org/pdfs/PIP_blogging_data.pdf`

[3] `http://techcentralstation.com/011105B.html`

[4] `http://www.cyberjournalist.net/news/001461.php`

[5] `http://politics.blogpulse.com/04_11_04/politics.html`

[6] `http://www.powerlineblog.com/archives/007760.php`

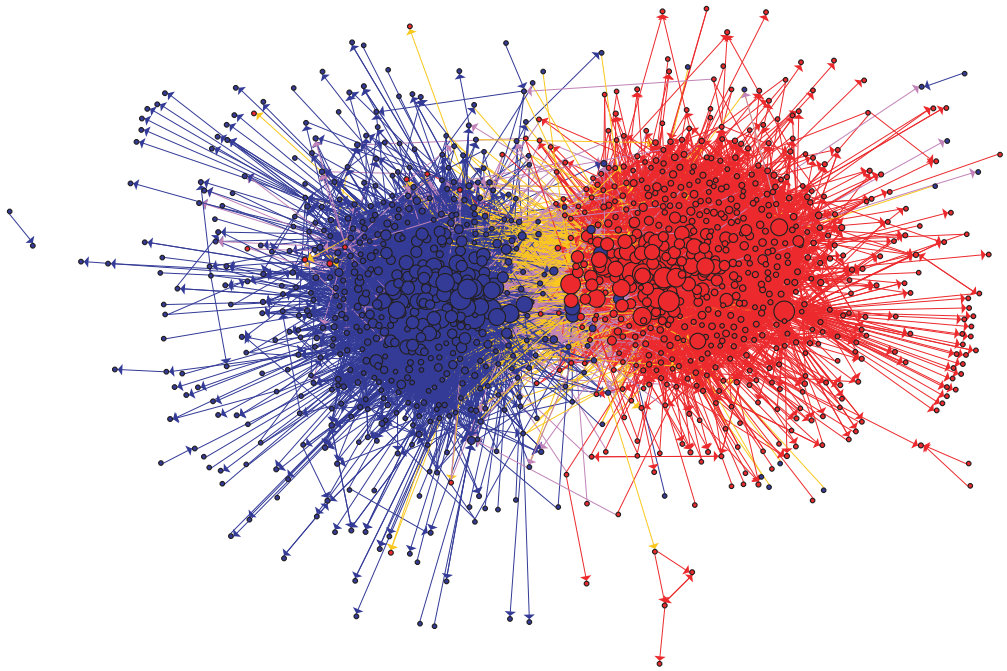


Figure 1: Community structure of political blogs (expanded set), shown using utilizing the GUESS visualization and analysis tool[2]. The colors reflect political orientation, red for conservative, and blue for liberal. Orange links go from liberal to conservative, and purple ones from conservative to liberal. The size of each blog reflects the number of other blogs that link to it.

Because of bloggers' ability to identify and frame breaking news, many mainstream media sources keep a close eye on the best known political blogs. A number of mainstream news sources have started to discuss and even to host blogs. In an online survey asking editors, reporters, columnists and publishers to each list the "top 3" blogs they read, Drezner and Farrell [4] identified a short list of dominant "A-list" blogs. Just 10 of the most popular blogs accounted for over half the blogs on the journalists' lists. They also found that, besides capturing most of the attention of the mainstream media, the most popular political blogs also get a disproportionate number of links from other blogs. Shirky [12] observed the same effect for blogs in general and Hindman et al. [7] found it to hold for political websites focusing on various issues.

While these previous studies focused on the inequality of citation links for political blogs overall, there has been comparatively little study of subcommunities of political blogs. In the context of political websites, Hindman et al. [7] noted that, for example, those dealing with the issue of abortion, gun control, and the death penalties, contain subcommunities of opposing views. In the case of the pro-choice and pro-life web communities, an earlier study [1] found pro-life websites to be more densely linked than pro-choice ones. In a study of a sample of the blogosphere, Herring et al.[6] discovered densely interlinked (non-political) blog communities focusing on the topics of Catholicism and homeschooling, as well as a core network of A-list blogs, some of them political.

Recently, Butts and Cross [3] studied the response in the structure of networks of political blogs to polling data and election campaign events. In another political blog study, Welsch [15] gathered a single-day snapshot of the network neighborhoods of Atrios, a popular liberal blog, and Instapundit, a popular conservative blog. He found the Instapundit neighborhood to include many more blogs than the Atrios one, and observed no overlap in the URLs cited between the two neighborhoods. The lack of overlap in liberal and conservative interests has previously been observed in purchases of political books on Amazon.com [8]. This brings about the question of whether we are witnessing a cyberbalkanization [11, 13] of the Internet, where the proliferation of specialized online news sources allows people with different political leanings to be exposed only to information in agreement with their previously held views. Yale law professor Jack Balkin provides a counter-argument[7] by pointing out that such segregation is unlikely in the blogosphere because bloggers systematically comment on each other, even if only to voice disagreement.

In this paper we address both hypotheses by examining in a systematic way the linking patterns and discussion topics of political bloggers. In doing so, we not only measure the degree of interaction between liberal and conservative blogs, but also uncover differences in the structure of the two communities. Our data set includes the posts of 40 A-list blogs over the period of two months preceding the U.S. Presidential Election of 2004. We also study a large network of over 1,000 political blogs based on a single day snapshot that includes blogrolls (the list of links to other blogs frequently found in sidebars), and so presents a more static picture of a broader blogosphere.

From both samples we find that liberal and conservative blogs did indeed have different lists of favorite news sources,

[7] `http://balkin.blogspot.com/2004_01_18_balkin_archive.html#107480769112109137`

people, and topics to discuss, although they occasionally overlapped in their discussion of news articles and events. The division between liberals and conservatives was further reflected in the linking pattern between the blogs, with a great majority of the links remaining internal to either liberal or conservative communities. Even more interestingly, we found differences in the behavior of the two communities, with conservative blogs linking to a greater number of blogs and with greater frequency. These differences in linking behavior were not drastic, and we can not speculate how much they correlated, if at all, with the eventual outcome of the election. They were nonetheless interesting, and we believe they show an insightful glimpse into the online political discourse leading up to the election.

2. METHODOLOGY

In order to get a representative view of the liberal and conservative blog communities, we cast our nets wide and gathered a single day's snapshot of over a thousand political blogs. Since we also wanted to do a careful study of the heart of the political blogosphere, we then analyzed the posts for the two months preceding the election with a smaller set of 40 influential blogs.

2.1 Calling all political blogs

We gathered a large set of political blog URLs by downloading listings of political blogs from several online weblog directories, including eTalkingHead, BlogCatalog, CampaignLine, and Blogarama. The directories had surprisingly little overlap, and occasionally listed conflicting categories for a single blog, even within the same directory. We did not gather the URLs of libertarian, independent, or moderate blogs, which were far fewer in number. We attempted to retrieve a single, 'front' page for each blog on February 8, 2005. From this set of pages, we counted up all citations to political weblogs not on our original list. The approximately thirty blogs that were cited seventeen or more times by our original set of political blogs, but were not listed in one of the directories, were categorized manually based on posts and blogrolls and added to the set. We retrieved pages for these additional blogs on February 22, 2005. Neither the directory labels, which often rely on self-reported or automated categorizations, nor our manual labels, are 100% accurate. However, since we are considering the aggregate behavior of well over 1,000 blogs, a few mislabeled ones will not affect our results significantly.

The set we attempted to retrieve initially was surprisingly balanced. There were 1494 blogs in total, 759 liberal and 735 conservative. Of these, we retrieved pages that were at least 8KB in size for 676 liberal and 659 conservative blogs. Some of the blogs which were not retrievable either no longer existed, or had moved to a different location. When looking at the front page of a blog we did not make a distinction between blog references made in blogrolls (blogroll links) from those made in posts (post citations). This had the disadvantage of not differentiating between blogs that were actively mentioned in a post on that day, from blogroll links that remain static over many weeks [9]. Since posts usually contain sparse references to other blogs, and blogrolls usually contain dozens of blogs, we assumed that the network obtained by crawling the front page of each blog would strongly reflect blogroll links. 479 blogs had blogrolls through `blogrolling.com`, while many others simply maintained a list of links to their favorite blogs.

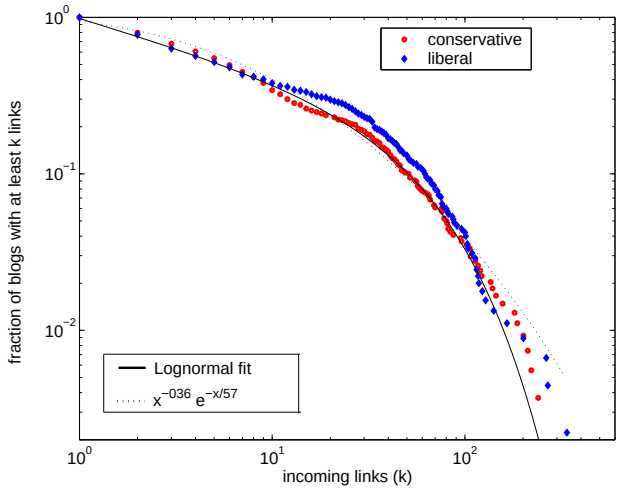


Figure 2: Cumulative distribution of incoming links for political blogs, separated by category. Both a lognormal and a power-law with an exponential cutoff provide good fits.

We constructed a citation network by identifying whether a URL present on the page of one blog references another political blog. We called a link found anywhere on a blog's page, a "page link" to distinguish it from a "post citation", a link to another blog that occurs strictly within a post. Figure 1 shows the unmistakable division between the liberal and conservative political (blogo)spheres. In fact, 91% of the links originating within either the conservative or liberal communities stay within that community. An effect that may not be as apparent from the visualization is that even though we started with a balanced set of blogs, conservative blogs show a greater tendency to link. 84% of conservative blogs link to at least one other blog, and 82% receive a link. In contrast, 74% of liberal blogs link to another blog, while only 67% are linked to by another blog. So overall, we see a slightly higher tendency for conservative blogs to link. Liberal blogs linked to 13.6 blogs on average, while conservative blogs linked to an average of 15.1, and this difference is almost entirely due to the higher proportion of liberal blogs with no links at all.

Although liberal blogs may not link as generously on average, the most popular liberal blogs, Daily Kos and Eschaton (atrios.blogspot.com), had 338 and 264 links from our single-day snapshot of political blogs. This is on par with the 277 links received by the most linked to conservative blog Instapundit. Figure 2 shows that, as is common in nearly every large subset of sites on the web [7, 10], the distribution of inlinks is highly uneven, with a few blogs of either persuasion having over a hundred incoming links, while hundreds of blogs have just one or two.

A small handful of blogs command a significant fraction of the attention, and it is these blogs that we will be analyzing in more detail in the following sections. We will return to a descriptive analysis of the larger set of blogs in Section 3.5.

Liberal						Conservative					
r	c	posts	blog	l_L	l_R	r	c	posts	blog	l_L	l_R
1	10053	1114	dailykos.com	292	46	2	8438	924	powerlineblog.com	26	195
6	6452	580	talkingpointsmemo.com	242	22	3	7813	740	instapundit.com	43	234
7	5468	945	atrios.blogspot.com	230	39	8	5298	682	littlegreenfootballs.com	10	171
9	4830	502	washingtonmonthly.com	165	36	11	3297	66	hughhewitt.com	11	146
18	2764	409	wonkette.com	83	30	12	3226	494	andrewsullivan.com	59	86
24	2277	211	juancole.com	149	16	13	3220	701	captainsquartersblog.com	5	117
30	1675	550	yglesias.typepad.com	104	24	14	3186	801	wizbangblog.com	14	125
33	1621	429	crookedtimber.org	81	19	16	2781	398	indcjournal.com	6	60
41	1365	348	mydd.com	107	8	17	2773	341	michellemalkin.com	10	191
45	1289	512	oliverwillis.com	97	20	19	2596	1027	blogsforbush.com	4	208
48	1268	767	blog.johnkerry.com	21	2	25	2259	87	allahpundit.com	2	37
49	1257	607	pandagon.net	118	5	26	2156	100	belmontclub.blogspot.com	3	93
54	1191	949	talkleft.com	126	15	27	1944	50	realclearpolitics.com	13	104
55	1142	345	digbysblog.blogspot.com	115	3	28	1882	633	volokh.com	27	80
56	1141	722	politicalwire.com	87	16	35	1570	510	timblair.spleenville.com	7	80
59	1077	470	j-bradford-delong.net	98	11	37	1523	428	windsofchange.net	16	65
66	1002	722	prospect.org/weblog	102	11	38	1512	595	vodkapundit.com	9	97
68	991	1653	americablog.blogspot.com	64	5	40	1468	446	rogerlsimon.com	6	74
74	947	582	theleftcoaster.com	78	4	42	1364	899	deanesmay.com	8	79
77	851	115	jameswolcott.com	74	6	44	1310	580	mypetjawa.mu.nu	0	51

Table 1: The top 20 liberal and conservative blogs by post citation count (c) and overall rank (r) according to BlogPulse data (October - November 2004). The two right columns show for comparison how many liberal (l_L) and conservative (l_R) blogs from the larger set linked to the blog in February 2005. Also included are the number of posts in our data set for each weblog.

2.2 Weblog selection

In order to perform an in-depth and balanced analysis, we decided to work with the top 20 conservative leaning and top 20 liberal leaning blogs, which we identified as follows. First we used page link counts to cull the top 100 or so conservative blogs and the top 100 liberal blogs from the larger list of 1494 political weblogs. Next, we used BlogPulse's index (`www.blogpulse.com/search`) of weblog posts to count up citations to this list of approximately 200 weblogs during the months of September and October 2004. Table 1 shows the citation count and overall rank among all weblogs for the top 20 lists.

It is interesting to note that the top 20 conservative leaning weblogs fall within the overall top 44 most cited weblogs for this time period. In contrast, the top 20 liberal leaning blogs fall within the overall top 77. This is evidence that bloggers in general link to conservative blogs more than to liberal blogs.

These lists have some notable omissions. For example, we chose not to include drudgereport.com, which had received 7813 blog post citations during Sep. - Oct. 2004, because of its unusual format and primary function as a mainstream news filter. We also omitted democraticunderground.com, which is principally a message board and secondarily a weblog.

In Table 1 we have also included the number of page links from the larger set of liberal and conservative blogs in February of 2005 to the top ranked blogs. The page link counts serve both to validate the popularity of the blogs and to show that the writings of a few bloggers, like Andrew Sullivan and Wonkette have across the board appeal, while most others are read almost exclusively by either the right or the left. It is immediately apparent that page link counts do not produce exactly the same rankings as post citations do. In fact, several blogs that are highly ranked according to page link counts, such as scrappleface.com, nationalreview.com/thecorner, thismodernworld.com, and outsidethebeltway.com do not make the top 20 lists (just barely missing, however). One factor is that the page links used for the rankings were collected in February 2005, while the post citation counts were tallied in the pre-election months. Even more importantly, as we have already argued, the blogrolls represented in the page link counts may be somewhat stale. Case in point are the 39 links to allahpundit.com and 23 links to blog.johnkerry.com found in February 2005, even though the former's author had retired in December 2004, and the latter had not been updated since October 2004. On the other hand, weblog rankings via post citation count are sensitive to the time period used for the calculation. The top 20 lists vary from month to month, although the very top most cited weblogs tend to remain the same.

In Table 2 we compare the top 10 blogs derived using BlogPulse data with the ranks assigned by several different ranking sites. TheTruthLaidBear and Technorati rely on link counts, while SiteMeter ranks according to traffic to the blog. Different approaches have different drawbacks. SiteMeter can only rank blogs that use its traffic meter and cannot differentiate unique visitors. Link-reliant rankings are affected by the freshness of the links and run the hazard of being manipulated by inventive bloggers. Despite the differences in ranking algorithms, we observe that the top 10 blogs in each list overlap significantly amongst themselves and with our top 20 rankings.

Because different ranking approaches can produce slightly different sets of top-ranking blogs, we checked that our results are robust with respect to the particular selection of the top blogs. We ran the set of analysis presented in Section 3 with variations in the set of weblogs, replacing some

	Technorati	SiteMeter	TheTruthLaidBear	BlogPulse
1	Instapundit	Daily Kos	Instapundit	Daily Kos
2	Daily Kos	Instapundit	Daily Kos	Power Line
3	Eschaton	Eschaton	Power Line	Instapundit
4	Little Green Footballs	Little Green Footballs	Little Green Footballs	Talking Points Memo
5	Andrew Sullivan	Power Line	Michelle Malkin	Eschaton
6	Wonkette	Wonkette	Talking Points Memo	Little Green Footballs
7	Power Line	Smirking Chimp	Eschaton	Washington Monthly
8	Volokh Conspiracy	Michelle Malkin	Captain's Quarters	Hugh Hewitt
9	Michelle Malkin	Blog for America	Volokh Conspiracy	Andrew Sullivan
10	Lileks	Lileks	Wizbang	Captain's Quarters

Table 2: Different methods produce different rankings, but the overlap in the top rated blogs is high. (Feb. 23, 2005 for Technorati, SiteMeter and TruthLaidBear; October - November 2004 for BlogPulse)

of the lower ranked top 20 with weblogs lower in the list. We found that qualitatively our results remain the same.

2.3 Data collection

We created a corpus of weblog posts from the top 20 conservative leaning blogs and the top 20 liberal leaning blogs. Table 1 includes the number of posts we harvested for each weblog for the time period 8/29/04 - 11/15/04. In all, we collected 12,470 posts from the left leaning set of blogs and 10,414 posts from the right leaning set.

We used BlogPulse's collection system to harvest posts. The strength of BlogPulse's collection system is that it is able to crawl weblog pages and segment the pages into individual posts. BlogPulse currently monitors over 5.5 million weblogs and indexes 450K weblog posts per day. Its coverage falls short of other comprehensive weblog search systems such as Technorati and PubSub because of its requirement that it be able to identify individual full-content posts. Segmentation is a trivial task for a weblog with a full-content feed (assuming that the feed can be automatically discovered). The task is trickier for weblogs with partial content feeds or no feed. In this case, BlogPulse uses a model-based wrapper learner to extract individual posts. Search over this index of weblog posts is publicly available at `http://www.blogpulse.com` [5].

From a corpus of individual posts, it is straightforward to analyze just the active interblog citation behavior, while discarding links that are automatically displayed on the blog page, such as those contained in blogrolls. Blogrolls, as we discussed previously may grow stale over time, and we are interested in tracking active discussions occurring between the A-list blogs.

Another advantage of creating a corpus from posts is that, apart from errors in segmentation, there is no duplicate content in the corpus. In contrast, a weblog spider that crawls a snapshot of a weblog at regular intervals, or whenever the weblog updates, will collect snapshots of overlapping content. The resulting data collection would not lend itself to producing accurate analytics.

3. ANALYSIS

3.1 Strength of community

We contrasted the citation behavior in the posts of the top 20 liberal and top 20 conservative blogs. During the two months covered by our analysis, the top 20 liberal bloggers published 12,470 posts, compared to 10,414 for the conservatives. We then counted the number of posts in which each blog cited another blog. If a blog was cited more than once within the same post, the link was not double-counted. We found that liberal blogs cited one another 1511 times, compared to conservatives who cited one another 2110 times. Cross citing accounted for only 15% of the links, with liberals citing conservatives 247 times, and conservatives citing liberals 312 times. The interesting result is that even though the conservatives had 16% fewer posts, they posted 40% more links to one another, linking at a rate of 0.20 links per post, compared to just 0.12 for liberal blogs.

We further found that the citations were concentrated among a smaller subset of the top 20 liberal blogs, but were relatively more distributed among the conservative blogs. Our observations are illustrated in Figure 3. We start out with a connected network of 20 blogs of each kind, with the conservative network having 278 directed internal edges, the liberal one having 218, and 210 cross-edges between them (Figure 3A). Next we remove any edges that are not sufficiently reciprocated (have fewer than 5 citations in either or both directions). This leaves 40 bidirected edges within the conservative network and 25 for the liberal one, with only 3 reciprocated edges between them (Figure 3B). Now if we further require that the total number of citations between two blogs be at least 25, then the communities separate completely, and the liberals are left with only 12 edges while the conservatives have 23 (Figure 3C). Through these visualizations, we see that right-leaning blogs have a denser structure of strong connections than the left, although liberal blogs do have a few exceptionally strong reciprocated connections.

3.2 Varied conversations

The denser linking pattern of conservatives raises the question of whether the conservative bloggers had a more uniform voice than the liberal ones did. Conservative television programs and conservative talk radio have often been perceived by the left to be acting as an echo chamber or "noise machine" for Republican talking points. Of course, conservatives have accused liberals of being an echo chamber themselves.

We looked for an echo chamber effect by measuring the similarity in content for each pair of blogs, using a cosine similarity measure over the URLs and phrases found in their posts. In the context of URLs, once we remove from our analysis links to political blogs, liberals and conservatives have about the same similarity within their communities, and lower similarity between communities. Because we are

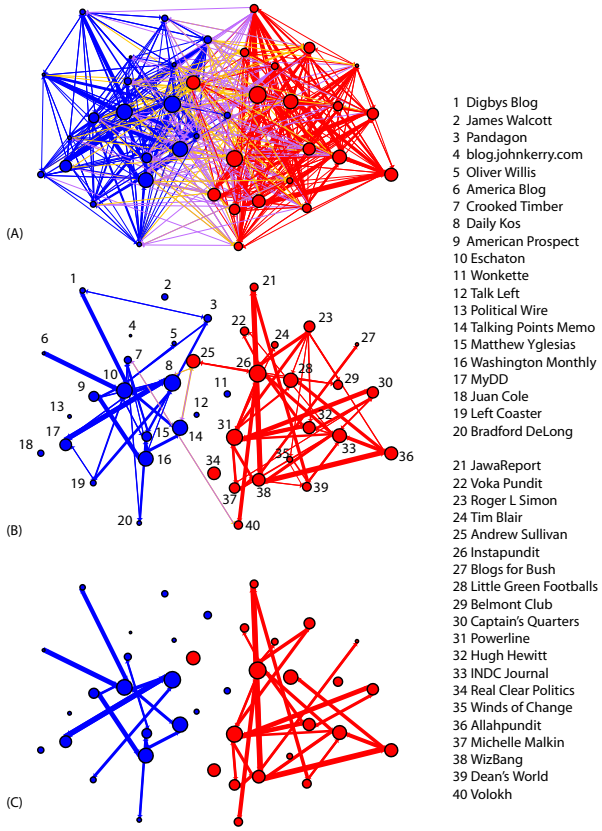


Figure 3: Aggregate blog citation behavior prior to the 2004 election. Color corresponds to political orientation, size reflects the number of citations received from the top 40 blogs, and line thickness reflects the number of citations between two blogs. (A) All directed edges are shown. (B) Edges having fewer than 5 citations in either or both directions are removed. (C) Edges having fewer than 25 combined citations are removed.

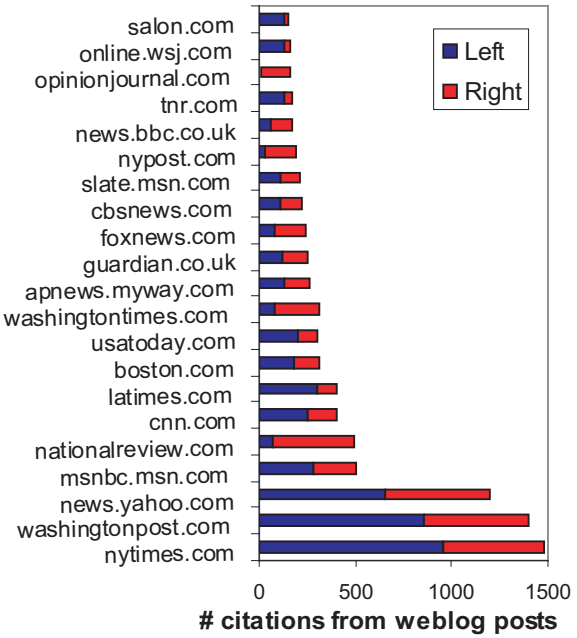


Figure 4: Most linked to news sources by the top 20 conservative and top 20 liberal blogs during 8/29/2004 - 11/15/2004.

excluding direct citations between the blogs (where the conservative blogs have a clear lead), the links capture echoing of some external sources directly, but the echoing between the blogs only indirectly. To measure textual similarity between blogs, we identified phrases that are most informative with respect to a background model of term frequencies in weblog data.[14]. Here we saw again a tendency, albeit a weaker one, for the same phrases to be used within communities than between them.

These results suggest that although conservative bloggers tend to more actively comment on one another's posts, this behavior is not accompanied by a greater uniformity in other online content they link to. Rather, we see both communities acting as mild echo chambers by frequently discussing separate sets of web pages and news items.

3.3 Interaction with mainstream media

Even more common than links to other blogs are links to news articles. Overall, the 20 left leaning bloggers cited the media 6,762 times, while the top 20 right leaning bloggers cited media 6,364, or about once every other post on average.

Figure 4 shows the most popular online news sites, and the proportion of liberal and conservative blogs linking to them within the top 20 liberal and the top 20 conservative blogs. As our analysis of the home pages of the larger set of political blogs will show in Section 3.5, we find that Fox News and the National Review receive the majority of their links from the conservative weblogs, while Salon receives over 86% of its links from liberal blogs.

Within the set of top political blogs, we also find that the NY Post, the WSJ Opinion Journal and the Washington Times receive the large majority of their links from right leaning blogs, while the LA Times, the New Republic and the Wall Street Journal are predominantly linked to by left leaning blogs. The remaining top-linked media sources are fairly evenly cited by the left and the right.

The actual news article citation behavior of the A-List political bloggers further differentiates the media sources attended to by bloggers on opposite sides of the political spectrum. Drilling down, here are the top news articles cited by left leaning bloggers:

1. CBS News poll of uncommitted voters shows Kerry winning 43% to 28%
2. Sun Times article: Bob Novak predicts that George Bush will retreat from Iraq if reelected
3. CBS News article on forged memos
4. New York Daily News article on Osama Bin Laden videotope, "gift" for the President
5. Time Magazine poll: Bush opens double-digit lead on post convention bounce

In contrast, the top news articles cited by right leaning bloggers are:

1. CBS News article on forged memos
2. Time Magazine poll: Bush opens double-digit lead on post convention bounce
3. National Review article refuting missing explosives case
4. ABC News article refuting missing explosives case
5. Washington Post article on Kerry's proposal to compromise with Iran on nuclear technology.

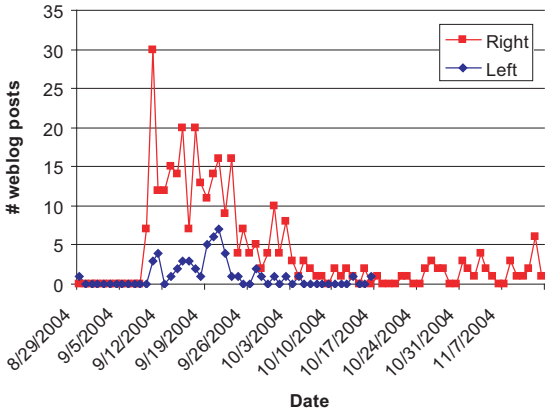


Figure 5: Time series chart: # posts discussing the CBS forged documents, right vs. left.

A time series chart further shows how quickly and strongly conservative bloggers responded to forged CBS documents (Figure 5). The conservative bloggers saw Dan Rather's report as an attempt by the left to discredit President Bush. They acted quickly to debunk the report, with the charge led by PowerLine and seconded by Wizbangblog and others. In contrast, the pick-up among liberal bloggers occurred later, with lower volume. The most vocal left leaning bloggers on the subject were TalkLeft and AMERICAblog.

3.4 Occurrences of names of political figures

Figure 6 shows the relative number of mentions of the most cited political figures in our set of top 40 political weblogs. This chart excludes George W. Bush and John Kerry, with 8071 and 8011 mentions, respectively, in order to improve readability. Interestingly enough, the right leaning bloggers account for 59% of mentions of John Kerry, while the left account for 53% of mentions of George Bush.

The chart shows that some political figures are the focus of attention of primarily one side of the political spectrum. For example, the following figures are cited by name predominantly by the right: Dan Rather, Michael Moore, Yasser Arafat and Terry McAuliffe. On the other hand, the left leaning bloggers account for most mentions of: Donald Rumsfeld, Colin Powell, Zell Miller and Tim Russert.

Notice the overall pattern: Democrats are the ones more often cited by right-leaning bloggers, while Republicans are more often mentioned by left-leaning bloggers. (While Zell Miller is officially a Democrat, he spoke at the Republican Convention and has been outspokenly anti-Kerry). These

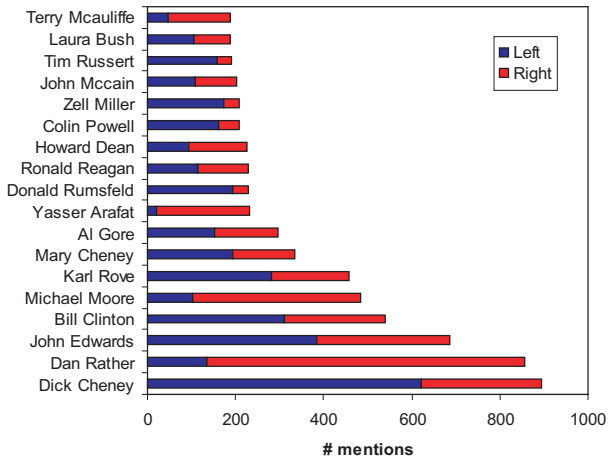


Figure 6: Mentions of political figures in liberal vs. conservative weblogs (excludes George W. Bush and John Kerry)

statistics indicate that our A-list political bloggers, like mainstream journalists (and like most of us) support their positions by criticizing those of the political figures they dislike. An interesting topic for further study would be to compare how balanced bloggers' presentation of the facts are compared with that of mainstream media journalists.

To create this data, we ran a person name extractor over the sets of left-leaning weblog posts and right-leaning weblog posts and counted up occurrences of the same name. We excluded the names of weblog authors for our top 40 weblogs. We then grouped together by hand different variants of the same name. The person name extractor has a recall of about 90%, so it is to be expected that one to three major political figures are missing from the ranking.

3.5 Back to the Greater Political Blogosphere

In addition to the A-list political blogs, there are hundreds of other blogs in the political blogosphere. We compared their interests in mainstream media based on the links extracted from their pages, and found these closely mirrored those of the top 40 blogs. Fox news and the National Review receive 89% and 92% of their links, respectively, from conservatives, while Salon receives 91% of its links from liberal blogs. The New York Times and Washington Post are the most balanced, with the Washington Post being slightly favored by conservatives (55 to 40), while the NYT was almost even, with 5 more liberals than conservatives linking to it. We noticed that some news sources often linked to by bloggers on their sidebars are cited relatively less frequently in blog posts. For example, Salon receives about 60% as many blogroll links as does the New York Times. But it only receives 11% as many citations from blog posts, suggesting that its articles are not discussed as often by A-list bloggers.

Links to non-blog and non-news sites reveal other preferences that the two categories of blogs have. In what follows, there are two numbers next to each site, the first being the number of links from liberal blogs, the second being the

number of links from conservatives. Conservatives get their daily cartoon fix at conservative cartoonist blogs 'Day by Day'(1,48) and 'Cox and Forkum'(1,83), while liberals get their laughs at the Onion (50,14). Liberals organized around thereisnocrisis.com (85,2) to defend social security against the plans of President Bush. They also link to MoveOn.org (56,9), and lend an ear to Michael Moore (44,10). The conservatives pay attention to the Middle East Media Research Institute (3,37), and listen to Sean Hannity (2,39) and Rush Limbuagh (3,29). They link to the GOP site (0,37) and are influenced by conservative think-tanks: the Heritage Foundation (1,33) and the Cato Institute (3,25).

4. CONCLUSIONS AND FUTURE WORK

In our study we witnessed a divided blogosphere: liberals and conservatives linking primarily within their separate communities, with far fewer cross-links exchanged between them. This division extended into their discussions, with liberal and conservative blogs focusing on different news, topics, and political figures. An interesting pattern that emerged was that conservative bloggers were more likely to link to other blogs: primarily other conservative blogs, but also some liberal ones. But while the conservative blogosphere was more densely linked, we did not detect a greater uniformity in the news and topics discussed by conservatives.

There are still many questions that we would like to explore regarding the political blogging community. Some would involve extending our current methods, and others would take us further into the intricate behavior of the blogosphere. As a refinement of our current approach, we would like to differentiate between blogs written by a single author and those written by multiple contributors, since a few blogs in our two top 20 lists have (or had) multiple authors. We would also like to track the spread of news and ideas through the communities, and identify whether the linking patterns in the network affect the speed and range of the spread. For example, we observed that the CBS document discussion was much more active in the conservative blogosphere. Was it because of stronger interaction patterns of the conservatives, or did the liberals simply not want to discuss it?

Finally, there is the matter of the 'other' political blogs, those calling themselves 'independent' or 'moderate'. Very few popped up on our radar, with none making our top 20 lists. At the very least, however, we could see whether they act as bridges between the liberal and conservative communities, or if they form their own community which may be just as isolated. Will they grow in number and start gaining in popularity as the divisive 2004 U.S. Presidential election fades from memory? Any change in the balance and interaction of the political blogosphere makes for an interesting subject of study.

5. ACKNOWLEDGMENTS

We would like to thank Eytan Adar and Drago Radev for insightful discussions and TJ Giuli and Marita Silverstein for helpful comments and suggestions. We would also like to thank the BlogPulse team, especially Matt Hurst and Mark Reed for their vital technical contributions, Sundar Kadayam for his vision and guidance, and Sue MacDonald for her inspired editorial analyses of the political blogosphere.

6. REFERENCES

[1] L. A. Adamic. The small world web. In *Proceedings of the 3rd European Conf. on Digital Libraries*, volume 1696 of *Lecture notes in Computer Science*, pages 443–452. Springer, 1999.

[2] E. Adar. Guess: The graph exploration system. `http://www.hpl.hp.com/research/idl/projects/guess/guess.html`, 2005.

[3] C. Butts and R. Cross. Blogging for votes: An examination of the interaction between weblogs and the electoral process. Sunbelt XXV presentation.

[4] D. W. Drezner and H. Farrell. The power and politics of blogs. `http://www.danieldrezner.com/research/blogpaperfinal.pdf`, 2004.

[5] N. Glance, M. Hurst, and T. Tomokiyo. BlogPulse: Automated trend discovery for weblogs. In *WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[6] S. C. Herring, I. Kouper, J. C. Paolillo, and L. A. Scheidt. Conversations in the blogosphere: An analysis "from the bottom up". In *HICSS-38*. Springer, 2005.

[7] M. Hindman, K. Tsioutsiouliklis, and J. A. Johnson. "googlearchy": How a few heavily-linked sites dominate politics on the web. `www.princeton.edu/~mhindman/googlearchy--hindman.pdf`, 2004.

[8] V. Krebs. The social life of books, visualizing communities of interest via purchase patterns on the www. `http://www.orgnet.com/booknet.html`, 2004.

[9] C. Marlow. Audience, structure and authority in the weblog community. In *International Communication Association Conference*, New Orleans, LA, 2004. `http://web.media.mit.edu/~cameron/cv/pubs/04-01.pdf`.

[10] D. M. Pennock, G. W. Flake, S. Lawrence, E. J. Glover, , and C. L. Giles. Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences (PNAS)*, 99(8):5207–5211, 2002.

[11] R. Putnam. *Bowling Alone*. Simon and Schuster, New York, 2000.

[12] C. Shirky. Power laws, weblogs and inequality. `http://shirky.com/writings/powerlaw_weblog.html`, 2003.

[13] C. Sunstein. *republic.com*. Princeton University Press, Princeton, 2001.

[14] T. Tomokiyo and M. Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL Workshop on Multiword Expressions*, 2003.

[15] P. Welsch. Revolutionary vanguard or echo chamber? political blogs and the mainstream media. Sunbelt XXV presentation, 2005.

Graph Building as a Mining Activity: Finding Links in the Small

Antonio Badia
abadia@louisville.edu

Mehmed Kantardzic
mmkant01@louisville.edu

Computer Engineering and Computer Science department
University of Louisville
Louisville KY 40292

ABSTRACT

Many analysis of data proceed by building a graph out of the data set and then using *social network theory* and similar tools on the result. However, there is no theory concerning the construction of the graph itself, even though this is a very important process. In this paper, we attempt to provide a framework in which the graph building process is formalized and studied. We show the parameters (choices) involved in constructing a graph from raw data, and propose some new ways to combine and analyze the data. We also argue the importance of this approach in several domain applications, including criminal/terrorist investigations.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Link Analysis

Keywords

network, link analysis, graph algorithms

1. INTRODUCTION

Recently, it has become quite fashionable to analyze data by building a graph out of a set of observations and applying tools and techniques of graph theory (in this paper, we will use the terms *graph* and *network* interchangeably). The recent emphasis in *social network theory* ([23]) is one example of this approach. Usually, the graph techniques receive all the attention, while the process of creating the graph is left to the skills of the analyst ([16]). Indeed, very little thought has been paid in the research literature to the process of building the graph. However, it has become clear that this process is of fundamental importance, since the results obtained by posterior analysis depend crucially on the graph being built ([9]). At the same time, graph building includes several judgments about what the data means, and how it is better organized. In the end, it is perfectly possible to build more than one graph from the same data set. Given the importance of the process, it seems overdue to carry out a (semi)formal analysis of graph building, in order to clarify the process and the results.

In this paper, we propose a framework for building graphs out of observational data. To the best of the author's knowledge, this is the first such framework proposed in the research literature. In the next section we introduce the framework for graph building. After discussing the overall process, we discuss how to define and choose nodes, and in subsection 2.2, we discuss several measures of connectivity, including two new measures of our own. The whole process is put together in subsection 2.3. We then propose in section 3 several ways to analyze the resulting graph, paying special attention to the discovery of *relevant* connections. We discuss some related work in section 4. Finally, since we are reporting on ongoing work, we close with a discussion of open issues and future work.

2. A FRAMEWORK FOR GRAPH DESIGN

We consider the process of building graphs from data as composed of two steps. The first step is *going from raw data* to some suitable, uniform format; the second one is *building the graph* from such a representation. This division allows us to tackle separately preparatory work (analyzing the domain, dealing with the data -cleaning, standardizing, etc.) and the creation of the graph per se. It also allows us to deal with the fact that data initially may come in many different formats, and it is impossible to anticipate all possible ways in which data is reported

When data is collected from a given domain, usually a *conceptualization* step takes place first (sometimes implicitly). The domain is analyzed and carved out into *objects* or *entities* that are grouped into *classes* or *types*; these objects have *attributes* or *properties*, and are related to each other through some *relationships*. These basic building blocks are used by most conceptual models, like the Entity-Relationship models ([21]) or class diagrams in UML ([18]), as well as in more advanced knowledge representation formalisms. The latter are quite similar, although they add more constructs in order to capture more meaning ([4]); however, here we restrict ourselves to the basics in order to be as general as possible. As an example, a domain could be seen as consisting of the classes *People*, *Places*, *Financial Institutions*, and the relationships *Travel* between *People* and *Places* and *has-*

account between *People* and *Financial Institution*. *People* could have attributes *name*, *age* and *country-of-origin*.

Usually, a researcher has a conceptualization in mind, even if it is never made explicit. When the graph is built, it is assume that (some of) the types will provide the domain for the nodes of the graph (that is, objects will be mapped to nodes), and that the relationships among them will be the basis for computing the link (based on some *measure*, see later). In the simplest case, we are dealing with only one type of data, and one binary relation defined on this type, possibly with some attributes (of the objects and/or the relationship). For the rest of this paper we restrict ourselves to this case, which we call the *homogeneous* case, to simplify our presentation; however, most of what we state here applies to the full or *heterogeneous* case. As an example of a homogeneous domain, we have a domain of *People* and the relationship *send-message* defined on it, with the domain playing the roles of *sender* and *recipient*. The information in a homogeneous model can be represented in a *table*, in the sense of relation, as in relational databases. The attributes on the schema of such table are: two attributes, each one for the role that the objects play in the relationship; plus any attributes of the objects, plus any attributes of the relationship[1]. Considering the data in a table allows us to introduce some uniformity, as raw data may be given in a large number of different formats. As an example that we use throughout the paper, let a table *Emails* have attributes `From, To, Date, Content`, as in Figure 1. This is meant to be a list of emails, each row being a particular email. The attributes `From` and `To` are the two roles of the underlying domain *People* (implicit); the rest are attributes of the relationship. The attribute `Content` is the text of the email; in order not to introduce complex text analysis issues ([3, 2]), in our example we simplify the content to a category.

Thus, this first step involves filling the individual attributes of each row from one or more observational datum. The mapping from the datum to the attributes can be quite complex, and may involve what is usually called cleansing, standardization, and other steps ([14]). We are fully aware of the importance of such details, and are currently developing formalizations for the whole process; however, such developments are outside the scope of the present paper.

The second step is *going from the table to a graph*. Once the data is in a table, one still needs to decide how to create the graph. This is the step in which we concentrate our efforts, and which is described in detail in the rest of the paper. We divide the process into three steps: deciding what constitutes a node, deciding what constitutes a link, and putting the resulting graph together.

As stated above, there is usually an implicit choice for what constitutes a node and what is a link that comes from the conceptualization. Our basic intuitions are: that such choices *should be made explicit*, as they determine the network and its interpretation, and constitute the basic parameters of the network construction process; and that *it is desirable to have as many degrees of freedom in determining what is a node and what a link*, since the process of creating a network out of a basic set of observations is guided by the analysis and goals that a user has in mind. In particular, *it is possible (and sometimes desirable) to build multiple graphs from the same conceptualization (data)*. Finally, we believe that by making the process explicit and as free as possible, new ways of analyzing the graph may be defined. We present some examples in later sections.

[1]It could be argues that the general or heterogeneous case would not lend itself to this simplified account; note, however, that we do not require anything from this table. In particular, we do not require that it be in any *normal form*; one could think of a table for the heterogeneous case as the *universal relation* considered in database theory ([22]).

2.1 What's in a Node?

Let C be a collection of facts with attributes $\mathcal{A} = A_1, \ldots, A_n$ (i.e. a table with schema $\mathcal{A}$). Each element of C (i.e. tuple) represents a *fact*. In the following, π is the relational projection operator (recall that for a table $\mathcal{A}$ with schema $A_1, \ldots, A_n$, the projection of $\mathcal{A}$ onto $B_1, \ldots, B_m$ - with $\{B_1, \ldots, B_m\} \subseteq \{A_1, \ldots, A_n\}$- is the table that results from taking each tuple (row) in $\mathcal{A}$ and "narrowing it down" to the attributes $B_1, \ldots, B_m$). Also, $|A|$ denotes the cardinality of set A.

A set of attributes $\mathcal{O} \subset \mathcal{A}$, with maximum cardinality $n-1$, is called an *object definition*. Given object definition $\mathcal{O} = A_i, \ldots, A_j$ $(1 \leq i, j \leq n)$, the set of objects of type $\mathcal{O}$ in C (in symbols, $ext(\mathcal{O}, C)$) is simply $\pi_{\mathcal{O}}(C)$. Thus, each unique combination of values for the chosen attributes identifies an object.

Emails			
From	**To**	**Date**	**Content**
P_1	P_2	1	A
P_2	P_5	1	A
P_3	P_4	2	C
P_1	P_3	3	B
P_2	P_3	3	B
P_4	P_1	4	F
P_1	P_2	5	C
P_3	P_6	5	C
P_2	P_5	6	D
P_3	P_4	1	A
P_4	P_1	1	C
P_1	P_3	2	C
P_2	P_3	5	C
P_5	P_3	2	C
P_3	P_4	1	A
P_3	P_3	5	C
P_4	P_2	1	A

Figure 1: Set of raw data

EXAMPLE 1. *In table* Emails*, a user may choose* From *to be the object definition (in which case the set of objects of type* `From` *is the set of all people sending messages:* $\{P_1, P_2, P_3, P_4, P_5\}$*), which means the analysis is centered in who sends the messages. Or, the user could choose* `Date` *(in which case the set of objects of type* `Date` *is the set of all dates in which messages were sent:* $\{1, 2, 3, 4, 5, 6\}$*), which means the analysis is centered on time. Choosing* `From, Date` *as a pair (in which case the set of objects of type* `From, Date` *is the set of all pairs (s,d) where s is a person, d is a date and s sent a message in date d:* $\{(P_1, 1), (P_2, 1), (P_3, 2), (P_1, 3), (P_2, 3), (P_4, 4), (P_1, 5), (P_3, 5), (P_2, 6), (P_3, 1), (P_4, 1), (P_1, 2), (P_2, 5), (P_5, 2)\}$*) means that the analysis is centered on who sends the messages when.*

Note that what are choosing, basically, is the underlying domain. Thus, as a special case, we allow two attributes specifying roles in a domain to be chosen together as in (`From` ∪ `To`), to denote the whole underlying domain of persons. Note that this is different from choosing `From, To` as the object definition, since in this case we get *pairs* of persons to be represented as a node.

Given object $o = (a_i, \ldots, a_j) \in ext(\mathcal{O}, C)$, the *support set* of o, in symbols $S(o)$, is defined as $\{t \in C \mid \pi_{\mathcal{O}}(t) = o\}$. In relational parlance, we will group the relation by the attributes in $\mathcal{O}$; the resulting groups are the support sets. Despite the name, repetitions are allowed.

EXAMPLE 2. **(Continuation)** *If the user chooses* `From` *as the object definition, then the support set of a given object (say,* <Jim Jones>*) is the set of all email messages sent by Jim Jones. If the user chooses* {`From, Date`} *as the object definition, then the support set of a given object (say,* <Jim Jones,Sept-15-2003>*) is the set of all email send by that particular person on that particular date. Continuing with our previous example, the support set of P_1 is the set of all tuples that have the value P_1 in attribute* `From`*:* $\{(P_1, P_2, 1, A), (P_1, P_3, 3, B), (P_1, P_2, 5, C), (P_1, P_3, 2, C)\}$.

Given object definition $\mathcal{O}$, the rest of the attributes ($\mathcal{A} - \mathcal{O}$) is called the set of *dimensions of the object*. Among such dimensions, a set of attributes $\mathcal{L} \subseteq \mathcal{A} - \mathcal{O}$ is called the set of *link dimensions*. Note that $\mathcal{L} \cap \mathcal{O} = \emptyset$. The *link support set* of an object $o \in ext(\mathcal{O}, C)$ with dimensions $\mathcal{L}$, in symbols $S(\mathcal{L}, o)$, is defined as $\{\pi_{\mathcal{L}}(t) \mid t \in S(o)\}$.

EXAMPLE 3. **(Continuation)** *If a user chooses* `From` *as the object definition, then* {`To, ID, Date, Body`} *is the set of dimensions of the object. Any subset of this set could be the set of link dimensions. In particular,* {`Date`} *is one such set. Assume an object (sender) o_1 fixed. The link support set in this case is, the set of dates in which o_1 sent a message. If the link dimension set is* {`Date,To`}*, the link support set is the set of pairs (d,p) where p received a message from o_1 in date d. Thus, following with our example from table* Emails*, given the support set of P_1 as before, the link support set is* $\{(1, A), (3, B), (5, C), (2, C)\}$.

The effect of our definition is to allow any combination of attributes to become the basis for a node. No longer is the node only a conceptualization object; any combination of attributes can be thought of as a node. While objects (in this example, people) as nodes is a natural and intuitive view, there is no reason that analysis cannot center on some other perspective. In our example, by choosing `Date` we can carry out a temporal analysis; by choosing `Content`, we can carry out a content-centered analysis.

2.2 What's on a Link?

We believe that the process of determining whether two objects are connected can be considered akin to measuring how similar two objects are. Many *similarity measures* have been proposed over time, all of them based on what is common between two sets (i.e. their intersection). It is easy to see that all of the following are similarity measures for the set $ext(\mathcal{O}, C)$:

- $R(o_1, o_2) = |S(\mathcal{L}, o_1) \cap S(\mathcal{L}, o_2)|$ (raw measure).
- $D(o_1, o_2) = \frac{2|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{|S(\mathcal{L},o_1)| + |S(\mathcal{L},o_2)|}$ (Dice's coefficient).
- $J(o_1, o_2) = \frac{|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{|S(\mathcal{L},o_1) \cup S(\mathcal{L},o_2)|}$ (Jaccard's coefficient).
- $C(o_1, o_2) = \frac{|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{\sqrt{|S(\mathcal{L},o_1)| \times |S(\mathcal{L},o_2)|}}$ (cosine coefficient).
- $O(o_1, o_2) = \frac{|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{min(|S(\mathcal{L},o_1)|, |S(\mathcal{L},o_2)|)}$ (overlap coefficient).

All these measures are well known ([3]). Note that all of them depend on the intersection $S(\mathcal{L}, o_1) \cap S(\mathcal{L}, o_2)$. We call this intersection the *common set*. The first measure is simply the size of the common set. We believe that this measure is the simplest one possible, but is of interest in some cases and is implicitly used many times when building graphs. These measures have some well known properties:

LEMMA 2.1. *All the above (except the raw measure) are similarity measures:*

1. *For S = D, J, C, O, S(X,Y) = S(Y,X) and $S(X, Y) = 1$ iff $X = Y$.*
2. *For S = D, J, C, O, $0 \leq S(X, Y) \leq 1$.*

LEMMA 2.2. *For any X, Y, $C(X, Y) \leq (J(X, Y), D(X, Y)) \leq O(X, Y)$.*

All the above measures are *normalized* (i.e. return a value between 0 and 1), except the raw measure (hence the name). This is a nice property, that allows meaningful comparison among values. However, to determine a *threshold* at which two objects can definitively be considered linked (which is our ultimate goal) is hard. We propose two new, non-normalized measures that will allow us to get around this problem, based on the following intuition: two objects are connected if they have more in common than they have different (∞ denotes a symbol larger than any positive number):

$$S_1(o_1, o_2) = \begin{cases} \infty & \text{if } S(\mathcal{L}, o_1) = S(\mathcal{L}, o \\ \frac{|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{max(|S(\mathcal{L},o_1) - S(\mathcal{L},o_2)|, |S(\mathcal{L},o_2) - S(\mathcal{L},o_1)|)} & else \end{cases}$$

$$S_2(o_1, o_2) = \begin{cases} \infty & \text{if } S(\mathcal{L}, o_1) = S(\mathcal{L}, o_2) \\ \frac{|S(\mathcal{L},o_1) \cap S(\mathcal{L},o_2)|}{|S(\mathcal{L},o_1) - S(\mathcal{L},o_2)| + |S(\mathcal{L},o_2) - S(\mathcal{L},o_1)|} & else \end{cases}$$

How do these new measures behave? For starters, it is easy to see that

LEMMA 2.3. *For any X, Y, $S_1(X, Y) \geq S_2(X, Y)$*

The following is a property that connects our measures to the previous ones:

LEMMA 2.4. *When $0 \leq S_i(X, Y) \leq 1$, $J(X, Y) \leq S_i(X, Y)$, for $i = 1, 2$.*

While these measures are not normalized, we can make a virtue out of this as follows. For any measure, we can consider two objects not related when the value is 0 (since this implies that the common set is empty -that is, the objects have nothing in common); but to consider all objects related we need some arbitrary threshold r to be set for D, J, C, and O. For S_1 and S_2, though, a natural threshold emerges: 1. In effect, if $S_2(o_1, o_2) > 1$, this means that o_1 and o_2 have more in common than they have different; and if $S_1(o_1, o_2) > 1$, this means that o_1 and o_2 have more in common than any one of them has different from the other.

Thus, we consider o_1, o_2 *disconnected* when $S_i(o_1, o_2) = 0$; *connected* when $S_i(o_1, o_2) > 1$; and neither connected nor disconnected in all other cases ($i = 1, 2$).

Some further observations can be made about these measures (in the following, S_i is used to refer to both S_1 and S_2). First, when both $S(\mathcal{L}, o_1)$ and $S(\mathcal{L}, o_2)$ are small, or when both $S(\mathcal{L}, o_1)$ and $S(\mathcal{L}, o_2)$ are large, the connection depends solely on how much they have in common. In particular, when both support sets are *very large* (close to the size of the domain), it may be that their intersection is large enough simply because the sets are. Our original meaning was to focus on small support sets (overlooked by most mining algorithms), but it seems not to be easy to achieve this without putting some (arbitrary) bound on the size of the support set.

Second, when one of $S(\mathcal{L}, o_1)$ and $S(\mathcal{L}, o_2)$ is large and the other one is small, our definition will rule out a connection. To see this, note that the largest set cannot be more than double the size of the other. The reason is the following: assume $S(\mathcal{L}, o_1)$ is the small one, $|S(\mathcal{L}, o_1)| = r$ and $|S(\mathcal{L}, o_2)| = 2r+1$. The strongest connection is the one where $S(\mathcal{L}, o_1) \subseteq S(\mathcal{L}, o_2)$, since the cardinality of the intersection cannot be larger than the cardinality of the sets involved. But even then, $|S(\mathcal{L}, o_1) \cap S(\mathcal{L}, o_2)| = |S(\mathcal{L}, o_1)| = r$, while $|S(\mathcal{L}, o_2) - S(\mathcal{L}, o_1)| = r+1$. Hence, there cannot be a large differential on the size of the support sets. We consider this situation analogous to the *negative correlation* problem for association rules, in which what seems like a statistically significant overlap tends to be caused by the overall distribution of the data. Thus, we wish to avoid such cases, as they are suspect and likely not to carry any meaning. In particular, the case where one support set is contained in the other, i.e. $S(\mathcal{L}, o_1) \subseteq S(\mathcal{L}, o_2)$ is such that, if $S(\mathcal{L}, o_2)$ is large, we may not have a connection in our sense! Although this may seem counterintuitive, we do not claim that there is no relation between the objects in such a case; simply, that this is not the relation that we are interested in. Note that the connection is symmetric (i.e. $S_i(o_1, o_2)$ iff $S_i(o_2, o_1)$), while the containment case is not. Containment relations are suspect because of the negative correlation problem.

Third, the relationship of being connected has the remarkable property that it does not have many properties that would seem obvious. For instance, it is not the case that if $S_i(o_1, o_2)$ and $S_i(o_2, o_3)$, then $S_i(o_1, o_3)$. To see why, an easy counterexample is $|S(\mathcal{L}, o_1)| = |\mathcal{L}, S(o_2)| = |\mathcal{L}, S(o_3)| = n$, and $|S(\mathcal{L}, o_1) \cap S(\mathcal{L}, o_2)| = \frac{n}{2}+1$, and $|S(\mathcal{L}, o_2) \cap S(\mathcal{L}, o_3)| = \frac{n}{2} + 1$. Then, it is indeed the case that $S_1(o_1, o_2)$ and $S_1(o_2, o_3)$; however, $|\mathcal{L}, S(o_1) \cap S(\mathcal{L}, o_3)|$ may be as small as 2! Hence, when $n > 4$, the transitivity does not have to hold. Also, it is not the case that if $S_i(o_1, o_2)$ and $S_i(o_1, o_3)$, then $S_i(o_2, o_3)$. It is not obvious, then, how to extend the definition to the case of a set of objects, as opposed to a pair, except in the obvious way (i.e. objects $o_1, \ldots, o_l$ are strongly connected iff they are pairwise strongly connected -if $S_i(o_k, o_j)$, $1 \leq k \neq j \leq l$). This definition leads to very costly algorithms to compute, so we do not adopt it here (although it may be interesting on its own).

2.3 Building Graphs

As usual, a *graph* is a collection of objects (from now on called *nodes*) and of links (from now on called *edges*). Edges can be classified according to their *directionality* (they may be one-way or directed, or two-way or undirected) and their *character* (they may be positive -denoting closeness of two objects- or negative -denoting differences o distance between two objects). The above measures can be combined to determine not only whether a link between two objects exist, but also, if it exists, whether it is directional or not, and its character ([8, 19]).

Given a collection of facts C, a graph or network for C under measure f is a triple $(\mathcal{O}, \mathcal{L}, f)$, where $\mathcal{O}$ is an object definition, $\mathcal{L}$ is a link dimension for $\mathcal{O}$ and f is a similarity measure for $\mathcal{O}$ and $\mathcal{L}$ (from now on, we'll consider f to be one of the measures introduced earlier: R, D, J, C, O, S_1 or S_2). The nodes in this graph are the objects $o \in ext(\mathcal{O}, C)$, and the edges are the links created according to link measure f as follows: for all measures f but S_1 and S_2, a positive edge is created between objects o_1 and o_2 if $f(o_1, o_2) > 0$ (if graph is labeled, the edge can be given weight $f(o_1, o_2)$); and a negative edge is created if $f(o_1, o_2) = 0$. To create an edge from S_1 (or S_2) means: create a negative edge if $S_1(o_1, o_2)$ ($S_2(o_1, o_2)$) is less than 1; a positive edge (with weight i) if $S_1(o_1, o_2) = i > 1$ ($S_2(o_1, o_2) = i > 1$); and no edge if $S_1(o_1, o_2) = 1$ ($S_2(o_1, o_2) = 1$). An edge can be given direction by using more than one similarity measure and comparing the result: for instance, if $S_1(o_1, o_2) << S_2(o_1, o_2)$ the edge should be given direction from the object with smaller support set to the one with the larger; if $S_1(o_1, o_2) \approx S_2(o_1, o_2)$, the edge is bidirectional. Other conditions may also be applied to determine possible directionality. However, using more than one measure is necessary since by definition measures are symmetric, while a directional edge is not symmetric.

EXAMPLE 4. **(Continuation)** *Following with our example of table* Emails*, the user chose* `From` *as the object definition, and* {`Date, Content`} *as link dimensions, so the graph means: person o_1 and person o_2 are sending messages at the same (or close) time with the same (or similar) content. We can easily compute the links, their support sets and their value under each measure. We don't show the whole resulting graph, focusing on the link between nodes P_1 and P_2 and on our measures. The support sets are as follows: for P_1, {(1,A), (3,B), (5,C), (2,C)}; for P_2, {(1,A), (3,B), (6,D), (5,C)}. Then, S_1 is computed as the size of the intersection of such sets (which is 3, as the intersection is {(1,A), (3,B), (5,C)}) divided by the symmetric difference (which is 2, as there is one element in P_1 minus P_2, and one element in P_2 minus P_1), for a value of 1.5 (if we use S_2, we divide by the maximum of the two, which is 1, leading to a value of 3). Therefore, we conclude that this is a strong link.*

3. GRAPH ANALYSIS

Finally, there is the issue of what are interesting operations that can be defined on graphs. Clearly, our graphs are not different from other graphs; we were merely analyzing the process of graph creation. However, the analysis gives us the basis to define operations on graphs that are meaningful for data analysis. Here we concentrate on three tasks: the determination of when a link is interesting; the comparison of different graphs derived from the same table; and the process of *extracting interesting subgraphs* from a given one.

3.1 Signi£cant Connections

The task is to determine whether a connection (link) is interesting. We believe that it is impossible to formalize com-

pletely a notion of *interestingness* or *relevance*, since such a notion is bound to be context-dependent and somewhat subjective. We can attempt, however, to capture some aspects of the concept.

Our approach is based on the following intuition: the formal definitions introduced before captured what is usually known as the *confidence* of a link, i.e. the strength that we associate with it. In order to be interesting, a link should also be *statistically significant* (i.e. should not occur by chance), and should be *rare* (not frequent). These two ideas seem to be in contradiction, as traditionally statistical significance has been taken to mean high frequency. Here, however, we take it to be simply that the events in question are more related than they would be by chance. To implement that, we use a χ^2 test for categorical data. This allows us to consider low-support events, and disregard those that are likely to be *noise*.

We require events to be infrequent but to contribute significantly to a link. The intuition here is that rare events should barely appear in any given connection; if they appear often, that is worth noting. In our example, if messages of a certain type are rare (they occur very infrequently), but most of such messages happen to be between two people, that should call our attention. Note that this does not *per se* guarantee that the event is interesting -only the analyst can ultimately make that determination- but it certainly warrants some attention.

For a given link, we compute three measures: one to determine whether the link is strong, another one to compute statistical significance, and another one to determine rarity. We explain our approach with an example. We continue with the example of the link between P_1 and P_2, as derived from the data in Figure 1. We have already determined the link is strong. We then apply the χ^2 test as follows: the expectation matrix in Figure 2 is created. Note that the square corresponding to $\bar{P_1}, \bar{P_2}$ has the number of values *in the whole table* that are not in the support set of P_1 or in the support set of P_2 -in the example, this corresponds to {(4,F), (1,C)}. Then the traditional χ^2 test yields a value of 3.8, which means that the link is statistically significant ($p > 0.05$). Now we compute our third value, *rarity*. We use three measures for this: *Selective Rarity (SR)*, *local Rarity (LR)* and *Global Rarity (GR)*. We compute, for each element in the common set of the link, how many times the element appears in the whole dataset, and compare this number of the number of appearances in the common set under consideration. For instance, (1,A) occurs 5 times in the table, of which 2 are in the link between P_1 and P_2; there are two occurrences of (3,B) in the database, and both are in the link between P_1 and P_2; and (5,C) occurs 4 times in the whole database, 1 of those in the link between P_1 and P_2. We then compute, for each data point, the fraction of those two values as *number of occurrences in the link support set / total number of occurrences*. For (1,A), this value is $\frac{2}{5} = .4$; for (3,B), the value is $\frac{2}{2} = 1$ (the maximum possible value); for (5,C), the value is $\frac{1}{4} = .25$. SR chooses the maximum of all these values, which in our example is 1. This is as high as it can get, and therefore we deem the event interesting. LR takes the average instead of the maximum; so $avg(0.4, 1, 0.25) = .55$. This measure should be above 0.5, so we also deem the link interesting under this measure. Finally, our last check includes the fact that the element under consideration is indeed infrequent in the data. We notice that (1,A) occurs 5 times out of 17 observations, (3,B) occurs 2 times out of 17, while (5,C) occurs 4 times out of 17. Note that the minimum frequency is $\frac{1}{N}$, where N is the total number of facts under consideration (17, in this example). The GR of a value is defined as the ratio of this fraction to the SR of the value. For instance, (3,B) occurs infrequently (2/17) while its SR is as high as possible (1). These two facts combined indicate that we should consider this an interesting event: one that is rare and yet, happens frequently between P_1 and P_2.

	P_2	$\bar{P_2}$	Total
P_1	3	1	4
$\bar{P_1}$	1	2	3
Total	4	3	7

Figure 2: Expectation Matrix for P_1, P_2

We believe that this is of interest to law-enforcement, counter-terrorism and intelligence work because in these lines of work it may be necessary to detect small but potentially significant events. For instance, in a very large database of calls (say, with one million calls), a few of them (say, 10) mentioning nitrates and other fertilizer material that can be also used to make explosives may go unnoticed (since 10 calls out of one million may be simply noise). But if most of the calls happen to be between the same two people (say, 8 out of 10 calls were between two individuals), this is a fact that may be worth investigating. While there may be many such cases (small number of calls about a given subject), the tests introduced above should help weed out those that are likely noise and allow the analyst to focus on a few, potentially relevant leads.

3.2 Building Related Graphs

One of the central tenets of this approach is that, starting from the same body of facts, it is possible to build more than one graph. The question then is, given two different graphs G_1 and G_2 from the same set of facts C, is it possible to compare G_1 and G_2 or to combine them in some meaningful way? In some particular cases, this is indeed possible. Given a graph $G = (\mathcal{O}, \mathcal{L}, f)$, we define $N(G) = \mathcal{O}$, $E(G) = \mathcal{L}$. Consider two graphs G_1 and G_2 defined over the same C with the same f and the following situations:

- $N(G_1) = N(G_2)$, $E(G_1) \neq E(G_2)$. In this case we are considering the same objects, but examining their relationships under different aspects. It is possible to combine such graphs in a meaningful way, by constructing a graph with $N(G_1)$ and some combination of $E(G_1)$ and $E(G_2)$. That is, for $o_1, o_1 \in N(G_1)$, a link between them is the result of combining somehow the links between them in $E(G_1)$ and the links between then in $E(G_2)$; however, one must be careful about what the resulting graph means. Assume, for instance, that $N(G_1) = N(G_2) = \{From\}$ and $E(G_1) = \{Date\}$, while $E(G_2) = \{Body\}$. Then it is clearly possible to combine such graphs; however, the result obtained will be, regardless of the method chosen to carry out the combination of links, different from the graph obtained by setting $E(G) = \{Date, Body\}$. This is due to the fact that, when evaluating the support sets for G, we

require that two particular emails agree (under some measure) on both date and body. However, on G_1 we only require agreement on date, and on G_2 we only require agreement on the body. Even if later we combine the information, there is no way to enforce that agreement on date and body is realized for dates and bodies coming from the same emails. Of particular interest is the case $E(G_1) \subset E(G_2)$, in which are examining the same objects, progressively adding more information to the link; in this case we say that G_2 is a *link refinement* of G_1. Clearly, given C we can build a hierarchy of graphs $G_1,\ G_2, \ldots, G_n$ such that G_{i+1} is a link refinement of G_i $(1 \leq i < n)$. Graphs so obtained should be more and more *sparse* (less edges).

- $N(G_1) \subset N(G_2)$. In this case, we say that G_1 is a *node refinement* of G_2. For instance, assume $N(G_1) = \{From\}$, $N(G_2) = \{From, Date\}$. Clearly, the objects in G_1 can be grouped (by Date) so that they correspond to objects in G_2; a set of objects in G_1 correspond to an object in G_2. When $E(G_1) = E(G_2)$, we say that G_1 is a *proper node refinement* of G_2. We can see G_1 as giving the same analysis as G_2, only at a more fine-grained level. However, if $E(G_1) \neq E(G_2)$, the analysis may focus on different aspects, not just on different levels. As before, given C we can build a hierarchy of graphs $G_1,\ G_2, \ldots, G_n$ such that G_i is a proper node refinement of G_{i+1} $(1 \leq i < n)$. An especially interesting case is when $N(G_{i+1}) \subset N(G_i) \cup E(G_i)$ and $E(G_{i+1}) \subseteq E(G_i) - N(G_{i+1})$, i.e. some of the attributes in the link dimension are added to the object definition (and taken away from the new link dimension).
- $N(G_1) = E(G_2)$ and $N(G_2) = E(G_1)$. In this case we say that G_1 and G_2 are transposes of each other (if we represent the graphs as matrices, the resulting matrices are the transpose of each other).

The process of refinement is an interesting one, but we can also think of some quasi-inverse processes that are also relevant for analysis. Given graph G we call the graphs G_1, G_2 obtained by setting $N(G_1)$ and $N(G_1)$ to be a partition of $N(G)$ (that is, $N(G_1) \subset N(G)$, $N(G_2) \subset N(G)$, $N(G_1) \cap N(G_2) = \emptyset$ and $N(G_1) \cup N(G_2) = N(G)$) an *implosion* of G. The name comes from the fact that what was considered a node before is broken down into parts; this allows us to focus on different aspects of the object and see how much each one contributes to the overall connectivity.

Another useful graph transformation is to use a set of *constraints*, formulas that determine when links in G can be redrawn and how.

EXAMPLE 5. *Assume a graph* $G = (\{\texttt{From}, \texttt{To}\}, \{\texttt{Date}\}, f)$ *(we don't care for now which* f *we choose). Then a node* o_{12} *and a node* o_{34} *are linked if persons 1 and 3 are sending messages to persons 2 (for 1) and 4 (for 3) at (about the) same time (see figure 3). A semantically useful implosion would be to link nodes when person 1 and person 3 are the same (meaning: person 1 is sending messages to persons 2 and 4 at about the same time) (see figure 4). Note that this is different from the graph formed by* $(\{\texttt{From}\}, \{\texttt{To}, \texttt{Date}\}, f)$*! Such a graph would be formed if we use the constraint that person 2 and person 4 are the same. Another useful implosion would be based on the constraint that person 2 and person 3 are the same, in which case one gets the graph in figure 5, in which person 1 sends messages to person 2. which are then forwarded to person 3 at about the same time. Note that the connection is made based on the* same *messages being sent from 1 to 2 and then from 2 to 3 (i.e. forwarding!), and that this is different from a graph obtained using* $\{From\}$ *alone to characterize the object, as there is no link dimension and measure that will allow us to make a comparison across nodes (all comparisons are pairwise, between pairs of nodes).*

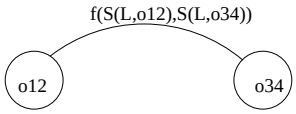


Figure 3: Initial graph

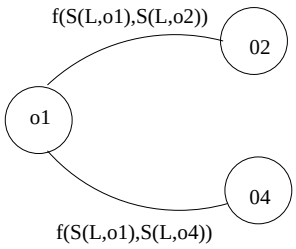


Figure 4: Initial graph, imploded

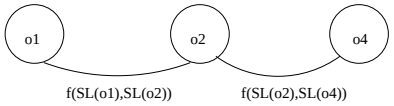


Figure 5: Initial graph, imploded in another way

3.3 Finding Interesting Subgraphs

Graphs can be *labeled* (i.e. symbols attached to their nodes or links to incorporate more domain semantics). It seems clear that one particular label that may be very meaningful is to *annotate each link with the common set that originated it.* That is, a link between objects o_1 and o_2 exists, first and foremost, due to $S(\mathcal{L}, o_1) \cap S(\mathcal{L}, o_2)$ (in any of the measures proposed) -this is what we called its common set. Assume now that we have a link l_1 between objects o_1 and o_2, and a link l_2 between objects o_2 and o_3. We would like to find out if objects o_1 and o_3 are related (and, if so, how). The connection can then be defined based on the underlying common sets for links l_1 and l_2; that is, the links may be composed only if their underlying common sets pass some test. Assume, for instance, that the link dimensions on this graph were $\{Date, Body\}$ (so that common sets are made up of pairs $< d, b >$, where d is the date of an email and b is its body). Then one can define an *interesting connection* between o_1 and o_3 to exists iff $S(\mathcal{L}, o_1, o_2) \cap S(\mathcal{L}, o_2, o_3) \neq \emptyset$ (note that the intersection here is computed on sets of pairs). This idea can be extended without difficulty to *paths* (or, to be technically accurate, to paths of length > 1). We can use this idea to derive a graph G' from a graph G as follows:

for each set of nodes $o_1, o_2, o_3 \in N(G)$ such that there is a link between o_1 and o_2 and a link between o_2 and o_3, compute the interesting connection. If one exists, create nodes (o_1, o_2) and (o_2, o_3) and a link between them. Note that different topologies in G given raise to nodes with different *labels* in G'. For instance, the example just given is one of a path; cycles give raise to other labeling, as do *stars* (i.e. graphs in which one central node o is connected to all the rest of the nodes $o_1, \ldots, o_n$, which are not connected among themselves). These three example topologies are given in Figure 6, for a simple graph with 3 nodes and 2 links, assuming both links are found interesting (the original graph G on the left, the resulting G' to its right). Given two arbitrary nodes o_1 and o_2 in graph G, the question *Is there a connection between o_1 and o_2, and if so, what is its nature?* can be answered as follows: if there is a path in the original graph G, compute G', G'', G''',... that is, compute the transitive closure of the graph under the interesting connection relationships. If a direct link is established between o_1 and o_2, then there is indeed a connection; the nature of the connection can be explained by looking at the successive common sets computed along the way.

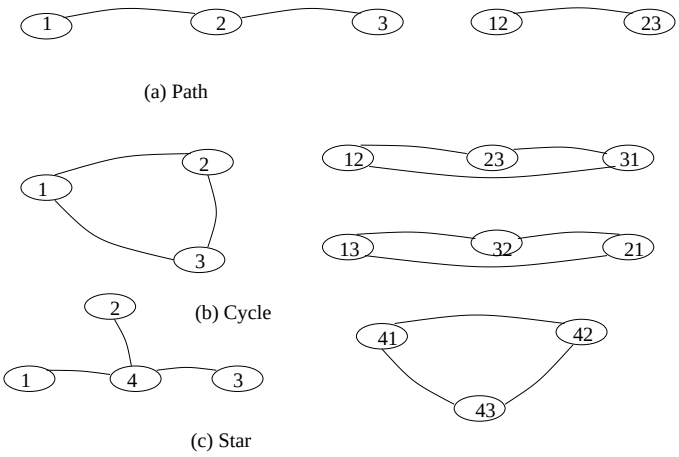


Figure 6: Interesting Connection Computation

Note that we can relax the idea of interesting connection to allow it to be based on a subset of the link support. If the original linking was based on {`To, Body`}, a and b were connected if they sent messages to the same person d about the same topic t, and b and c were connected if they sent messages to the same person d' about the same topic t'. We could be interested in connecting a and c if they sent messages out to the same person (i.e. $d = d'$), or if they sent messaged out about the same topic (i.e. $t = t'$). Of course, we could also be interested in them only if they sent out message to the same person about the same topic (i.e. if $d = d'$ *and* $t = t'$). Relaxing or tightening the constraints imposed on the connection, we can determine subgraphs of the original graph which are more or less tightly (and interestingly) connected. Continuing with law-enforcement example, once we have determined that persons a and b are talking about some suspicious substances a lot more than could be expected by chance, we may want to see who a and b talked to about the same subject, or about the same time that they were talking to each other (or both). Again, we insist that it is ultimately the analyst who must decide, based upon further analysis and context knowledge, whether an event is of interest. The purpose of our process is to allow the analyst to focus on a few facts of potential interest, among a large volume of data.

3.4 Preliminary Experiments

We have applied our approach to the Enron dataset ([1]). We have started by creating a table similar to the one used in our examples from the raw material (directories of emails). So far we have only created graphs for a subset of the whole collection. Our analysis uses people as nodes, and different graphs with different link dimensions have been created. One interesting fact is that, so far, the values for most links seem to have a Zipf distribution, suggesting that power laws apply to most combinations of attributes ([5]). As an example, figure 7 shows the top results when using `From` as node and `To, Date` as links.

Node: From
Link 1: To
Link 2: Date
Threshold: 6

FROM	Similarity measure	FROM
charles.alvarez@enron.com	77.00	adrian.woolcock@enron.com
martha.sumner-kenney@enron.com	51.00	maria.salazar@enron.com
donna.martens@enron.com	32.00	david.roensch@enron.com
brad.morse@enron.com	25.67	adrian.woolcock@enron.com
charles.alvarez@enron.com	25.67	brad.morse@enron.com
karim@karimrashid.com	19.88	info@david.se
ed.cattigan@enron.com	11.00	adam.overfield@enron.com
sophie.martin@enron.com	11.00	krzysztof.forycki@enron.com
hreasonen@velaw.com	8.00	lgodfrey@susmangodfrey.com
people@venturewire.com	8.00	alert@venturewire.com
keith.dziadek@enron.com	7.33	dan.dietrich@enron.com
khymberly.booth@enron.com	7.27	angie.collins@enron.com
stowsky@haas.berkeley.edu	7.00	hawe@haas.berkeley.edu
tracy.geaccone@enron.com	6.25	james.saunders@enron.com
john.allison@enron.com	6.17	cassi.wallace@enron.com
aaron@global2000.net	6.00	gwilliams@nyiso.com

Figure 7: Experiment Data

4. RELATED WORK

There is a considerable amount of work that can be regarded as somewhat relevant to this project; we focus here on research that is directly related to the work presented herein. There has been research on graph mining (see [7] for an overview); however, none of it analyzes the process of graph construction. There has also been research on determining when a relationship between two variables is interesting, usually in the context of association rule mining ([20, 11, 10, 12, 13, 6]). However, some of this work still assumes the typical support and confidence measures; since support excludes small sets, such work does not cover the same ground as ours. Most of the work derives from statistical concepts and therefore focuses on statistical relevance ([11]). Even on the context of graph mining, frequency is used to determine relevance ([15]). One exception is [6], which proposes to use Jacquard's coefficient, presenting its use in the context

of association rules, and defining a probabilistic algorithm to compute the measure very efficiently. Various notions of rarity are introduced to measure the interestingness of paths in bibliographical datasets in [17]. While experiments show promising results, better handling of noise, corruption and directionality of links is desirable. Messages with rare words are analyzed in [19] using *independent component analysis (ICA)*, when correlated words with "wrong" frequencies trigger detection of interesting messages. Other work has looked at connections in the context of Web searches ([9]).

5. CONCLUSION AND FURTHER RESEARCH

We have shown a formalization of the process of building graphs from raw data, an important process well due for an analysis. We have shown some graph manipulation that can be defined in the framework.

Since the research reported here is in its early stages, there are still several issues that must be dealt with. First and foremost is whether the framework is general enough. There are many ways in which an analyst may be interested in constructing a graph, possibly incorporating some type of constraints in the process. Further questions that still need to be answered include: how to accommodate facts of different importance; how to accommodate sets of facts that are not independent of each other; and how to accommodate uncertain/fuzzy facts. Also, more analysis is needed on how different but related graphs over the same data can be combined. Another issue is the presence and use of *time* and *space* dimensions (the attribute *Date* in our examples is one such dimension). Ideally, such dimensions capture more semantics and make possible more specialized analysis incorporating spatio-temporal reasoning. As an example, assume a database of disease occurrences. Two cases occurring on the same location and at the same time may warrant further investigation, while two cases at different locations will not be regarded in the same way. One interesting question raised by our experiments is whether any combination of attributes gives raise to a power law distribution ([5]). Finally, of course, is the issue of finding efficient algorithms to implement the proposed manipulations.

6. REFERENCES

[1] Enron data set. available at `http://www-2.cs.cmu.edu/ Enron/`.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 19999.

[3] R. Belew. *Finding Out About*. Cambridge University Press, 2000.

[4] A. Borgida. Knowledge representation, semantic modeling: Similarities and differences. In *Proceedings of the 9th International Conference on Entity-Relationship Approach*, 1990. Keynote address.

[5] S. Chakrabarty and C. Faloutsos. Graph structures in data mining. Tutorial in the SIGKDD 2004 Conference.

[6] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. Ullman, and C. Yang. Finding interesting associations without support pruning. In *Proceedings of the 16th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 2000.

[7] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[8] P. Domingos. Multi-relational record linkage. In *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining*. ACM Press, 2004.

[9] G. Hue and et al. Implicit link analysis for small web search. In *Proceedings of SIGIR-03*, 2003.

[10] F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining and with relative interestingness measure. In *Proceedings of the Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference (PADKK)*, number 1805 in Lecture Notes in Computer Science. Springer, 2000.

[11] S. Imberman, B. Domanski, and H. W. Thompson. Using dependency/association rules to find indications of computarized tomography in a head trauma dataset. *Artificial Intelligence in Medicine*, 26(1-2), September-October 2002. Elsevier.

[12] S. Jaroszewicz and D. Simovici. A general measure of rule interestingness. In *Proceedings of Principles of Data Mining and Knowledge Discovery, 5th European Conference (PKDD)*, number 2168 in Lecture Notes in Computer Science. Springer, 2001.

[13] S. Jaroszewicz and D. Simovici. Interestingness of frequent itemsets using bayesian networks as background knowledge. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2004.

[14] T. Johnson and T. Dasu. Data quality and data cleaning: An overview. Tutorial in the 2003 ACM SIGMOD Conference.

[15] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *ICDM*, pages 313–320, 2001.

[16] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. Technical report, Department of Computer Science, University of Minnesota, 2002.

[17] S. Lin and H. Chalupsky. Unsupervised link discovery in multi-relational data via rarity analysis. In *Proceedings of ICDM*, 2003.

[18] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.

[19] D. B. Skillicorn. Detecting related message traffic. In *Workshop on Link Analysis, Counterterrorism, and Privacy, SIAM ICDM*, 2004.

[20] P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (SIGKDD)*. ACM Press, 2002.

[21] B. Thalheim. *Entity-Relationship Modeling*. Springer-Verlag, 2000.

[22] A. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press, 1988.

[23] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

Mining Hidden Community in Heterogeneous Social Networks*

Deng Cai† Zheng Shao† Xiaofei He‡ Xifeng Yan† Jiawei Han†

† Department of Computer Science, University of Illinois at Urbana-Champaign

‡ Department of Computer Science, University of Chicago

ABSTRACT

Social network analysis has attracted much attention in recent years. Community mining is one of the major directions in social network analysis. Most of the existing methods on community mining assume that there is only one kind of relation in the network, and moreover, the mining results are independent of the users' needs or preferences. However, in reality, there exist multiple, heterogeneous social networks, each representing a particular kind of relationship, and each kind of relationship may play a distinct role in a particular task. Thus mining networks by assuming only one kind of relation may miss a lot of valuable hidden community information and may not be adaptable to the diverse information needs from different users.

In this paper, we systematically analyze the problem of mining hidden communities on heterogeneous social networks. Based on the observation that different relations have different importance with respect to a certain query, we propose a new method for learning an optimal linear combination of these relations which can best meet the user's expectation. With the obtained relation, better performance can be achieved for community mining. Our approach to social network analysis and community mining represents a major shift in methodology from the traditional one, a shift from single-network, user-independent analysis to multi-network, user-dependant, and query-based analysis. Experimental results on Iris data set and DBLP data set demonstrate the effectiveness of our method.

Keywords

Relation Extraction, Community Mining, Multi-relational Social Network Analysis, Graph Mining

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

LinkKDD'05, August 21, 2005, Chicago, IL, USA.

1. INTRODUCTION

With the fast growing Internet and the World Wide Web, Web communities and Web-based social networks are flourishing, and more and more research efforts have been put on Social Network Analysis (SNA) [16][21]. A social network is modeled by a graph, where the nodes represent individuals, and an edge between nodes indicates that a direct relationship between the individuals. Some typical problems in SNA include discovering groups of individuals sharing the same properties [18] and evaluating the importance of individuals [7]. In a typical social network, there always exist various relationships between individuals, such as friendships, business relationships, and common interest relationships.

Most of the existing algorithms on social network analysis assume that there is only one single social network, representing a relatively homogenous relationship (such as Web page linkage). In real social networks, there always exist various kinds of relations. Each relation can be treated as a **relation network**. Such kind of social network can be called *multi-relational social network* or *heterogeneous social network*, and in this paper the two terms will be used interchangeably depending on the context. These relations play different roles in different tasks. To find a community with certain properties, we first need to identify which relation plays an important role in such a community. Moreover, such relation might not exist explicitly, we might need to first discover such a hidden relation before finding the community on such a relation network.

Let us consider a simple example. In a typical human community, there may exist many relations: some people work at the same place; some share the same interests; some go to the same hospital, *etc.* Mathematically, this community can be characterized by a big graph in which the nodes represent people, and the edges evaluate their relation strength. Since there are different kinds of relations, the edges of this graph should be heterogeneous. For some tasks, we can also model this community using several homogeneous graphs. Each graph reflects one kind of relation. Suppose an infectious disease breaks out, and the government tries to find those most likely infected. Obviously, the existing relationships among people cannot play an equivalent role. It seems reasonable to assume that under such a situation the relation "*working at the same place*" or "*living together*" should play a critical role. The question becomes: "*How to select a relation that is most relevant to the disease spreading? Does there exist a hidden relation (based on the explicit relations) that best reveals the spread path of the disease?*"

These questions can be modeled mathematically as rela-

tion selection and extraction in *multi-relational* social network analysis. The problem of relation extraction can be simply stated as follows: *In a heterogeneous social network, based on some labeled examples (e.g., provided by a user as queries), how to evaluate the importance of different relations? Also, how to get a combination of the existing relations which can best match the relation of labeled examples?* In this paper, we propose an algorithm for relation extraction and selection. The basic idea of our algorithm is to *model this problem as an optimization problem*. Specifically, we characterize each relation by a graph with a *weight matrix*. Each element in the matrix reflects the relation strength between the two corresponding objects. Our algorithm aims at finding a linear combination of these weight matrices that can best approximate the weight matrix associated with the labeled examples. The obtained combination can better meet user's desire. Consequently, it leads to better performance on community mining.

It would be interesting to relate the relation extraction problem to the *feature extraction* problem [8] in machine learning. Feature extraction aims at the discovery of the intrinsic structure of a data set. This is similar to relation extraction, but used in different scenarios. Feature extraction is used when the objects have explicit vector representation, while relation extraction is used when only relationships between objects are available. Although feature extraction has been well studied in machine learning, there is still little research effort on relation extraction so far.

The rest of this paper is organized as follows. Section 2 presents background knowledge for relation extraction. Our algorithm for relation extraction is introduced and discussed in Section 3. The experimental results on the Iris data set and the DBLP data set are presented in Section 4. Our problem solving philosophy is discussed in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 8.

2. BACKGROUND

Social network analysis

Social network analysis (SNA) is the mapping and measuring of relationships and flows between people, groups, organizations, computers, or other information/knowledge processing objects. Social network analysis as a theme has been studied for years. The classic paper of Milgram [16] might be one of the first works on SNA. It estimates that every person in the world is only six "edges" away from every other. It sets the stage for investigations into social networks and algorithmic aspects of social networks. Many recent efforts try to leverage social networks for diverse purposes, such as expertise location [12], mining the network value of customers [7], and discovering shared interests [18].

Previous work in sociology and statistics has suffered from the lack of data and focused on very small networks, typically in the tens of individuals [21]. With the web growing, much potential social network data are available and a lot research efforts have been put on dealing with such data.

Schwartz and Wood mined social relationships from email logs [18]. The ReferralWeb project [12] is proposed to mine a social network from a wide variety of web data, and use it to help individuals find experts who could answer their questions. Adamic and Adar tried to discover the social interactions between people from the information on their homepages [1]. Agrawal et al. analyzed the social behavior of the people on the newsgroups [2]. Moreover, the web itself can be actually viewed as a large social network. The well-known link analysis algorithms, such as Google's PageRank [17] and Kleigberg's HITS algorithm [13], can be seen as social network analysis on the web.

Community mining

With the growth of the web, community mining has attracted increasing attention. A lot of work has been done at mining the implicit communities of web pages [10][14][9].

In principle, a community can be simply defined as a group of objects sharing some common properties. Community mining has many similar properties to the graph-cut problem. Kumar et al. used the bipartite graph concept to find the core of the community, and then expanded the core to get the desired community [14]. Flake et al. applied the maximum-flow and minimum-cut framework on the community mining [9]. The authority-and-hub idea was also used in the community mining [10][6].

Generally speaking, both social network analysis and community mining can be seen as graph mining. The community mining can be thought of as sub-graph identification. Previous work on graph mining can be found in [20]. Almost all the previous techniques on graph mining and community mining are based on a homogenous graph, *i.e.*, there is only one kind of relationship between the objects. However, in real social networks, there are always various kinds of relationships between the objects. To deal with this problem, we focus in this paper on multi-relational community mining.

Feature extraction

Feature extraction has received much attention in machine learning and data mining for its usefulness at classification and clustering. Feature extraction can be viewed as finding a linear combination of the original features that can better describe the intrinsic structure of the data set. Typical feature extraction methods include Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

Relation extraction, the problem raised in this paper, is fundamentally related to feature extraction. Given various relations of objects, we aim to find a linear combination of relations which can best reveal the intrinsic relationship between the objects with respect to the user's query. Unfortunately, the state-of-the-art feature extraction methods can hardly be applied to the relation extraction problem, since in relation extraction scenario the explicit vector representations of the objects are not available.

3. RELATION EXTRACTION

In this section, we begin with a detailed analysis of the relation extraction problem followed by two algorithms for two cases.

3.1 The Problem

A typical social network likely contains multiple relations. Different relations can be modeled by different graphs. These different graphs reflect the relationship of the objects from different views. For the problems of community mining, these different relation graphs can provide us with different communities.

As an example, the network in Figure 1 may form three different relations. Suppose a user requires the four colored

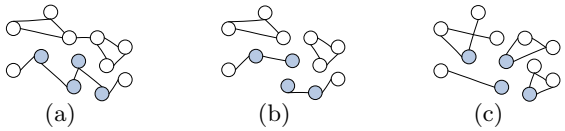


Figure 1: There are three relations in the network. The four colored objects are required to belong to the same community, according to a user query.

objects belong to the same community. Then we have:

1. Clearly, these three relations have different importance in reflecting the user's information need. As can be seen, the relation (a) is the most important one, and the relation (b) the second. The relation (c) can be seen as noise in reflecting the user's information need.

2. In the traditional social network analysis, people do not distinguish these relations. The different relations are equally treated. So, they are simply combined together for describing the structure between objects. Unfortunately, in this example, the relation (c) has a negative effect for this purpose. However, if we combine these relations according to their importance, the relation (c) can be easily excluded, and the relation (a) and (b) will be used to discover the community structure, which is consistent with the user's requirement.

3. In the above analysis, the relationship between two objects is considered as a boolean one. The problem becomes much harder if each edge is assigned with a real value weight which indicates to what degree the two objects are related to each other. In such situation, an optimal combination of these relations according to the user's information need cannot be easily obtained.

Different from Figure 1, a user might submit a more complex query in some situations. Take Figure 2 as another example. The relations in the network are the same as those in Figure 1. However, the user example (prior knowledge) changes. The two objects with lighter color and the two with darker color should belong to different communities. In this situation, the importance of these three relations changes. The relation (b) becomes the most important, and the relation (a) becomes the useless (and even negative) one.

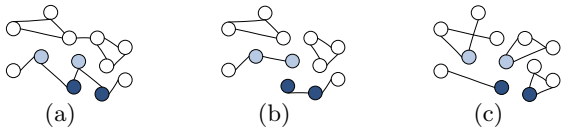


Figure 2: Among the three relations in the network, the two objects with lighter color and the two with darker color should belong to different communities, as user required.

As we can see, in multi-relational social network, community mining should be dependent on the user's example (or information need). A user's query can be very flexible. Since previous community mining techniques only focus on *single* relational network and are independent of the user's query, they cannot cope with such a complex situation.

In this paper, we focus on the relation extraction problem in multi-relational social network. The community mining based on the extracted relation graph is more likely to meet the user's information need. For relation extraction, it can be either linear or nonlinear. Due to the consideration that in real world applications it is almost impossible for a user to provide sufficient information, nonlinear techniques tend to be unstable and may cause over-fitting problems. Therefore, here we only focus on linear techniques.

This problem of relation extraction can be mathematically defined as follows. Given a set of objects and a set of relations which can be represented by a set of graphs $G_i(V, E_i)$, $i = 1, \ldots, n$, where n is the number of relations, V is the set of nodes (objects), and E_i is the set of edges with respect to the *i-th* relation. The weights on the edges can be naturally defined according to the relation strength of two objects. We use M_i to denote the weight matrix associated with G_i, $i = 1, \ldots, n$. Suppose there exists a hidden relation represented by a graph $\widehat{G}(V, \widehat{E})$, and $\widehat{M}$ denotes the weight matrix associated with $\widehat{G}$. Given a set of labeled objects $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$ and $\mathbf{y} = [y_1, \cdots, y_m]$ where y_j is the label of $\mathbf{x}_j$ (Such labeled objects indicate partial information of the hidden relation $\widehat{G}$), find a linear combination of the weight matrices which can give the best estimation of the hidden matrix $\widehat{M}$.

3.2 A Regression-Based Algorithm

The basic idea of our algorithm is trying to find an combined relation which makes the relationship between the intra-community examples as tight as possible and at the same time the relationship between the inter-community examples as loose as possible.

For each relation, we can normalize it to make the biggest strength (weight on the edge) be 1. Thus we construct the target relation between the labeled objects as follows:

$$\widetilde{M}_{ij} = \begin{cases} 1, & \text{example } i \text{ and example } j \text{ have} \\ & \text{the same label;} \\ 0, & \text{otherwise.} \end{cases}$$

where $\widetilde{M}$ is a $m \times m$ matrix and $\widetilde{M}_{ij}$ indicates the relationship between examples i and j. Once the target relation matrix is built, we aim at finding a linear combination of the existing relations to optimally approximate the target relation in the sense of L_2 norm. Sometimes, a user is uncertain if two objects belong to the same community and can only provide the possibility that two objects belong to the same community. In such case, we can define $\widetilde{M}$ as follows.

$$\widetilde{M}_{ij} = \text{Prob}(\mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same community})$$

Let $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T \in R^n$ denote the combination coefficients for different relations. The approximation problem can be characterized by solving the following optimization problem:

$$\mathbf{a}^{opt} = \arg\min_{\mathbf{a}} \|\widetilde{M} - \sum_{i=1}^{n} a_i M_i\|^2 \qquad (1)$$

This can be written as a vector form. Since the matrix $M_{m\times m}$ is symmetric, we can use a $m(m-1)/2$ dimensional vector $\mathbf{v}$ to represent it. The problem (1) is equivalent to:

$$\mathbf{a}^{opt} = \arg\min_{\mathbf{a}} \|\tilde{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{v}_i\|^2 \quad (2)$$

Equation (2) is actually a linear regression problem [11]. From this point of view, the relation extraction problem is interpreted as a prediction problem. Once the combination coefficients are computed, the hidden relation strength between any object pair can be predicted. There are many efficient algorithms in the literature to solve such a regression problem [4].

The objective function (2) models the relation extraction problem as an unconstrained linear regression problem. One of the advantages of the unconstrained linear regression is that, it has a close form solution and is easy to compute. However, researches on linear regression problem show that in many cases, such unconstrained least squares solution might not be a satisfactory solution and the coefficient shrinkage technique should be applied based on the following two reasons [11].

1. *Prediction accuracy*: The least-squares estimates often have low bias but large variance [11]. The overall relationship prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted relation strength, and hence may improve the overall relationship prediction accuracy.
2. *Interpretation*: With a large number of explicit (base) relation matrices and corresponding coefficients, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture", we are willing to sacrifice some of the small details.

Our technical report [5] provides an example to explain such consideration. This problem can be solved by using some coefficient shrinkage techniques [11].

Thus, for each relation network, we normalize all the weights on the edges in the range [0, 1]. And, we put a constraint $\sum_{i=1}^{n} a_i^2 \leq 1$ on the objective function (2). Finally, our algorithm tries to solve the following minimization problem,

$$\begin{aligned} \mathbf{a}^{opt} &= \arg\min_{\mathbf{a}} \|\tilde{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{v}_i\|^2 \\ &\text{subject to } \sum_{i=1}^{n} a_i^2 \leq 1 \end{aligned} \quad (3)$$

Such a constrained regression is called *Ridge Regression* [11] and can be solved by some numerical methods [4].

3.3 A MinCut-Based Algorithm

In the last subsection, we have presented a general method for exacting the hidden relation based on regression model. However, this method may fail when the examples provided by the user belong to only one community, which is referred to *single community issue* in the rest of this paper, please refer [5] for a more detailed example.

In order to deal with the *single community issue*, we need to take into account the weakest connection in the extracted relation. By graph theory, the value of the *minimum cut* on the graph can be used to evaluate the tightness of the graph.

Let G denote a weighted graph with weight matrix M. Let m denote the number of vertices. A cut on the graph is defined as a set of edges which separates the vertices into two disconnected groups denoted by A and B such that $A \cap B = \phi$ and $A \cup B = G$. Thus, the value of the cut is:

$$cut(G) = \sum_{i \in A} \sum_{j \in B} M(i,j)$$

It is easy to see that there are totally $2^m - 2$ different cuts. Let $cut_k(G) = (A_k, B_k)$ denote the k-th cut. The minimum cut is defined as:

$$mincut(G) = \min_k \{cut_k(G)\}$$

If a graph can be easily cut into two subgraphs, it has a small minimum cut value. As an extreme case, the minimum cut value of a disconnected graph is 0. Naturally, the optimal extracted relation graph should have a large minimum cut value. Thus, for single community issue, we try to extract the optimal relation graph by maximizing its minimum cut value.

Let G_i, $i = 1, \ldots, n$, denote the existing relation graphs defined only on the user query examples and M_i denote the corresponding weight matrices. Let $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T \in R^n$ denote the combination coefficients for different graphs. Thus $M = \sum_{i=1}^{n} a_i M_i$ is the weight matrix of the combined relation graph G. Let $mincut(G)$ denote the minimum cut value of G. Our objective function can be written as follows:

$$\mathbf{a}^{opt} = \arg\max_{\mathbf{a}} \{mincut(\sum_{i=1}^{n} a_i G_i)\} \quad (4)$$

In our problems, the number of examples provided by the user is usually small. That is, m is small, typically less than 10. Thus we can use linear programming techniques to solve the optimization problem (4) by the following derivation:

$$\begin{aligned} mincut(G) &= \min_{1 \leq k \leq 2^m - 2} \{cut_k(G)\} \\ &= \min_{1 \leq k \leq 2^m - 2} \{ \sum_{i \in A(k)} \sum_{j \in B(k)} M(i,j) \} \\ &= \min_{1 \leq k \leq 2^m - 2} \{ \sum_{i \in A(k)} \sum_{j \in B(k)} (\sum_{l=1}^{n} a_l M_l(i,j)) \} \\ &= \min_{1 \leq k \leq 2^m - 2} \{ \sum_{l=1}^{n} a_l (\sum_{i \in A(k)} \sum_{j \in B(k)} M_l(i,j)) \} \\ &= \min_{1 \leq k \leq 2^m - 2} \{ \sum_{l=1}^{n} a_l \cdot cut_k(G_l) \} \end{aligned}$$

Let $v = mincut(G)$. The optimization problem in Eq. (4) can be reduced to the following linear programming problem:

$$\begin{aligned} & \max \quad v \\ st. \quad & \sum_{l=1}^{n} a_l \cdot cut_k(G_l) - v \geq 0, \quad (1 \leq k \leq 2^m - 2) \quad (*) \\ & \sum_{l=1}^{n} a_l = 1 \\ & a_l \geq 0, \quad (1 \leq l \leq n) \end{aligned}$$

With the constraints (*), v is guaranteed to be the minimum cut value, and by maximizing v we can obtain the optimal combination coefficients a_i. The number of constraints

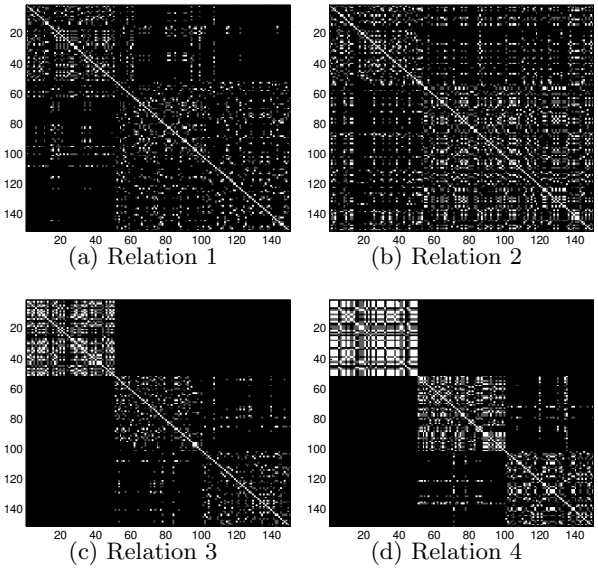


Figure 3: The four synthetic relation networks on Iris dataset

in this problem is $2^m - 2 + n + 1$, where m is the number of user-provided examples which is usually less than 10, and n is the number of existing relations. The above problem can be efficiently solved by linear programming techniques [3].

The proposed regression based algorithm and the MinCut based algorithm are used under different situations. When a user provides multiple community examples, regression-based algorithm can be used to find the best combination; when he provides single community examples, MinCut-based algorithm can be used.

4. EXPERIMENTS

In this section, we present our experimental study of the proposed relation extraction algorithm on the Iris and DBLP datasets. On the Iris dataset, we give quantitative results. On the DBLP dataset, some interesting examples are provided to show the effectiveness of our proposed algorithm.

4.1 Synthetic Relation Networks on Iris Data

In this section, we use the Iris data set to verify our algorithm. The iris dataset, popularly used for testing clustering and classification algorithms, is taken from UCI ML repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html). It contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. Each instance has four features, out of which it is known that F_3 (petal length) and F_4 (petal width) are more important for the underlying clusters.

For each feature F_r, we constructed a relation network $M_{r,ij}$ as follows:

$$M_{r,ij} = e^{-(x_i - x_j)^2} \tag{5}$$

Thus, the iris data can be viewed as a multi-relational social network with three hidden communities. The four relation matrices M_1, M_2, M_3, and M_4, constructed from the four features independently, were shown in Figure 3. The brightness reflects the relation strength between two objects.

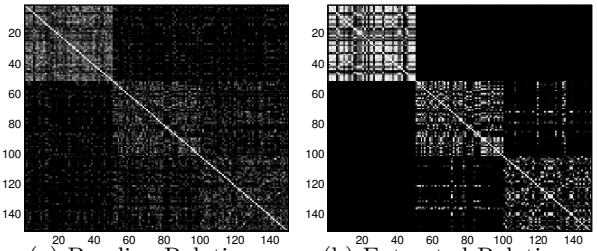


Figure 4: The baseline relation and extracted relation

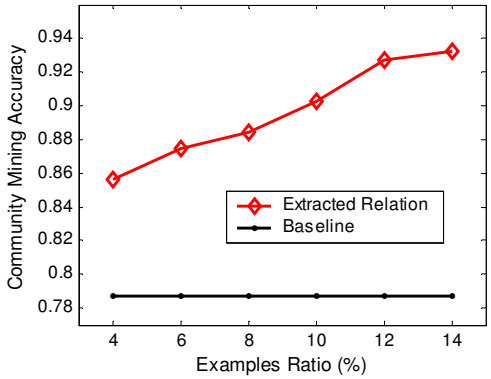


Figure 5: Community mining accuracy on the extracted relation

The following experiment was designed as given some labeled data as user query, using the regression based relation extraction algorithm described in Section 3.2 to extract the relation. Community mining was then applied on the extracted relation.

4.1.1 *Evaluation using community mining*

In this experiment, we applied the Normalized Cut algorithm [19][5] as our community mining algorithm.

The performance of community mining result is evaluated by comparing the obtained label of each object with the ground truth. Given an object $\mathbf{x}_i$, let r_i and s_i be the obtained community label and the ground truth, respectively. The accuracy AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{6}$$

where n is the total number of objects and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each community label r_i to the equivalent label from the ground truth. The best mapping can be found by using the Kuhn-Munkres algorithm [15].

4.1.2 *Results*

As described in Section 3, the extracted relation can be defined as,

$$M' = \sum_{i=1}^{4} a_i M_i \tag{7}$$

Traditional community mining algorithms are independent of the user-submitted query. Thus, the four relations are treated equally, *i.e.*, $a_i = 0.25$, $i = 1, \ldots, 4$. Community mining is then performed on this combined graph $M' = \sum_{i=1}^{4} 0.25 M_i$. We take this as the baseline. With the user's query, the relation extraction algorithm described in section 3 can be applied to extract a combined relation which respects the user's query. One can expect better performance with more labeled examples.

For a given number k, we randomly selected k examples for each class as the user query and extracted the optimal relation. The Normalized Cut algorithm was applied on the extracted relation to mine the communities and the accuracy was recorded. This process was repeated 100 times and the average performance was computed.

Figure 4 shows the baseline relation and the extracted relation. The extracted relation is obtained with 10% examples provided and the combination coefficients for the four base relations are 0, 0, 0.5948 and 0.8039, respectively. The community mining accuracy on extracted relation based on different examples ratio is shown in Figure 5.

This experiment shows that our algorithm can extract the optimal relation based on a few examples (labeled data). When some label information is available, it is always the case that a better relation can be extracted. In a multi-relational network, the user's information need can be extremely diverse. This particularly makes relation extraction an important pre-processing for social network analysis.

4.2 Mining Hidden Networks on the DBLP Data

In this part, we present our experimental results based on DBLP (Digital Bibliography & Library Project) data. The DBLP server (http://dblp.uni-trier.de/) provides bibliographic information on major computer science journals and proceedings. It indexes more than 500000 articles and more than 1000 different conferences (by May 2004).

Taking the authors in DBLP as objects, there naturally exist multiple relations between them. Authors publish paper in difference conferences. If we treat that authors publish paper(s) in the same conference as one kind of relation, these 1000 conferences provide us 1000 different relations. Given some examples (*e.g.*, a group of authors), our experiment is to study how to extract a new relation using such examples and find all the other groups in the relation. The extracted relation can be interpreted as the groups of authors that share a certain kind of similar interests.

4.2.1 Data preparation and graph generation

The DBLP server provides all the data in the XML format as well a simple DTD. We extracted the information of author, paper and conference. There are several points we should mention:

1. In the DBLP data, each author (researcher) is represented by a string. We do not distinguish two different authors carrying the same name. We believe that the duplicate names are only a tiny part of the whole dataset, thus will not be a problem in our experiment.

2. DBLP data does not provide the proceeding information for all the papers. We simply use the "key" attribute (usually in the format "conf/kdd/TsumotoT95a") to extract the conference identifier "KDD" and append with the "year" attribute. We assume that all papers in the same proceeding should have the same path name in the "key" attribute and the "year" attribute. This may not be true in all cases. However, we believe this is good enough for our experiments.

For each proceeding, we construct a graph with researchers as the nodes, which is called *proceeding graph* thereafter. If two researchers have paper(s) in this proceeding, the edge between the two corresponding nodes is set to 1. Otherwise, it is set to 0. For each conference, we add up the proceeding graphs of the same conference over years, which is called *conference graph* thereafter. Finally, we choose the top 70 conference graphs based on the number of distinct authors in that conference.

Every conference graph reflects the relationship between the researchers pertaining to a certain research area. Generally, if two researchers are connected by an edge in the conference graph, they may share the same research interests.

For each graph, we normalize the edge weight by dividing the maximum weight in the whole graph. The resulting weight has a range [0, 1]. The greater the weight is, the stronger the relation is.

4.2.2 Experiment results

In this experiment, we provide the system with some queries (some groups of researchers) to examine if our algorithm can capture the hidden relation between the researchers. The query examples are believed to belong to the same community, thus the MinCut based relation extraction algorithm was used.

Experiment 1. In the first case, there are two queries provided by the user.

1. Philip S. Yu, Rakesh Agrawal, Hans-Peter Kriegel, Padhraic Smyth, Bing Liu, Pedro Domingos.

2. Philip S. Yu, Rakesh Agrawal, Hans-Peter Kriegel, Hector Garcia-Molina, David J. DeWitt, Michael Stonebraker.

Both of the two queries contain 6 researchers. The first three researchers are the same in the two queries.

Table 1: Coefficients of different conference graphs for two queries (sorted on the coefficients)

Query 1		Query 2	
Conference	Coefficient	Conference	Coefficient
KDD	1	SIGMOD	0.528
		ICDE	0.262
		VLDB	0.210

Table 1 shows the coefficients of the extracted relation for the two queries. KDD is a data mining conference, and high weight on the KDD graph indicates the common interest on data mining. On the other hand, SIGMOD, VLDB and ICDE are three database conferences. High weights on these conference graphs indicate the common interest on database area. The extracted relation for query 1 has KDD graph with weighting 1, which tells us that the researchers in query

1 share common interest on data mining. For query 2, the extracted relation tells us those researchers share common interest on database.

Table 2: Researchers' activities in conferences

Researcher	KDD	ICDE	SIGMOD	VLDB
Philip S. Yu	7	15	10	11
Rakesh Agrawal	6	10	13	15
Hans-Peter Kriegel	7	9	11	8
Padhraic Smyth	10	1	0	0
Bing Liu	8	1	0	0
Pedro Domingos	8	0	2	0
Hector Garcia-Molina	0	15	12	12
David J. DeWitt	1	4	20	16
Michael Stonebraker	0	12	19	15

While we examine the publication of these researchers on these four conferences as listed in Table 2, we clearly see the extracted relation really captures the semantic relation between the researchers in the queries.

Furthermore, with the extracted relation graph, we applied the community mining algorithm *threshold cut* [5] and obtained the corresponding communities. For each query, we list one example community below:

1. Community for query 1: Alexander Tuzhilin, Bing Liu, Charu C. Aggarwal, Dennis Shasha, Eamonn J. Keogh, ...
2. Community for query 2: Alfons Kemper, Amr El Abbadi, Beng Chin Ooi, Bernhard Seeger, Christos Faloutsos, ...

Let us see what will happen if we only submit the first three names in one query. The extracted relation is shown in Table 3. The extracted relation really captures the two areas (data mining and dababase) in which these researchers are interested.

Table 3: Combined Coefficients

Conference Name	Coefficient
SIGMOD	0.302
KDD	0.279
ICDE	0.217
VLDB	0.202

Experiment 2. Let us try another example. The two queries are:

1. Pat Langley, Andrew W. Moore, Michael J. Pazzani, James P. Callan, Yiming Yang, Thomas G. Dietterich
2. Pat Langley, Andrew W. Moore, Michael J. Pazzani, Raymond T. Ng, Philip S. Yu

The extracted relations are shown in Table 4. And the activities of these researchers on these conference are shown in Table 5. In this case, we can draw the same conclusion as the first case. The relation extraction algorithm proposed in this paper can really extract the hidden relation from the user-provided examples.

From the above two experiments, one can see that data mining (KDD) is really an interdisciplinary area. Many researchers active in data mining are also active in database and machine learning. Our relation extraction algorithm captures the hidden relation among the researchers provide by users.

Table 4: Coefficients of different conference graphs for two queries (sorted on the coefficient)

Query 1		Query 2	
Conference	Coefficient	Conference	Coefficient
ICML	1	KDD	0.671
		ICML	0.329

Table 5: Researchers' activities in conferences

Researcher	ICML	KDD	SIGIR	SIGMOD
Pat Langley	11	4	0	0
Andrew W. Moore	10	3	0	0
Michael J. Pazzani	10	6	1	1
James P. Callan	3	0	10	1
Yiming Yang	5	1	8	0
Thomas G. Dietterich	12	1	0	0
Raymond T. Ng	0	8	0	7
Philip S. Yu	1	7	0	10

Experiment 3. In this case, four researchers mainly focus on different areas were submitted as the query. They are Avideh Zakhor, Lars Erik Holmquist, Elisa Bertino, and Makoto Sato. Based on the statistical information in the DBLP data, Avideh Zakhor focuses on ICIP, ISCAS, DCC, and CVPR; Lars Erik Holmquist on HUC, CHI, IWEC, and HCI; Elisa Bertino mostly on RIDE, ICDE, DBSEC, TIME, EDBT, and SIGMOD; whereas Makoto Sato mainly on VR.

Our algorithm extracted the hidden relation between them from the DBLP data. To our surprise, SIGGRAPH was selected as the hidden relation. When we carefully examined the DBLP statistics, we found that all these four researchers really showed up once in SIGGRAPH.

5. DISCUSSION

Since mining hidden communities in heterogeneous networks represents a promising research direction, there are many issues that need to be discussed. Here we focus on the problem solving philosophy.

First, one may *wonder the complexity at comprehension and combination of multiple social networks in the analysis.* We do agree that multiple social networks form complex, multiple, interrelated graphs, and with the massive amount of data mounting, it is challenging for anyone to grasp the whole picture of such dynamic, evolving social networks and work out a balanced combination of multiple networks for a particular user query. However, such multiple networks do exist, and it is inappropriate to blindly merge them into one since different networks plays different roles in particular queries, as shown in our experiments. Therefore, we believe that developing new multi-network mining algorithms to dynamically combine multiple relevant networks to form combined "virtual" networks based on user's example queries is a new and appropriate problem-solving methodology.

Second, since it is difficult for a user to comprehend the whole picture of numerous social networks, one may wonder *how a user is able to pose high-quality queries.* Based on our experience, although it is difficult for a user to compre-

hend the overall multiple networks, a user usually has good knowledge on a small set of examples (such as influential researchers, movie/sport stars, big companies, or popular commodities). Such firm grasp of a small set of examples is often sufficient to pose intelligent queries, learn additional facts, and form informative combined networks. This has been also demonstrated in our DBLP experiments.

Third, one may wonder *how to comprehend the answers returned from such a network analysis*. Since a derived hidden network is a weighted matrix as a combination of multiple existing networks, it is often difficult to understand the minor weight differences in the results. However, the real essence is at the new facts derived from such hidden networks and their associated rankings. This resembles Google-like keyword-based Web search. It is not so crucial to understand the derived Web linkage weighting and claim it is optimal. However, the return of quality rankings on the interesting results demonstrate its utility.

6. CONCLUSIONS

Different from most social network analysis studies, we assume that there exist multiple, heterogeneous social networks, and the sophisticated combinations of such heterogeneous social networks may generate important new relationships that may better fit user's information need. Therefore, our approach to social network analysis and community mining represents a major shift in methodology from the traditional one, a shift from single-network, user-independent analysis to multi-network, user-dependant, and query-based analysis. Our argument for such a shift is clear: multiple, heterogeneous social networks are ubiquitous in the real world and they usually *jointly* affect people's social activities.

Based on such a philosophy, we worked out a new methodology for relation extraction, and proposed two algorithms in different situations. With such query-dependent relation extraction and community mining, fine and subtle semantics are captured effectively. Our experimental results on Iris data set and DBLP data set demonstrate the effectiveness of our algorithm since it substantially improves prediction accuracy in comparison with the baseline approach and it convincingly discovers interesting relations and communities. Our discussion also shows it is expected that the query-based relation extraction and community mining would give rise to a lot of potential new applications in social network analysis.

7. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. Technical report, Xerox Parc, 2002.

[2] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of 12th International World Wide Web Conference*, 2003.

[3] M. Bazaraa, J. Jarvis, and H. Sherali. *Linear Programming and Network Flows*. Wiley, 3rd edition edition, 2004.

[4] A. Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[5] D. Cai, Z. Shao, X. He, X. Yan, and J. Han. Mining hidden community in heterogeneous social networks. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2005-2538, May, 2005.

[6] C. Chen and L. Carr. Trailblazing the literature of hypertext: Author co-citation analysis (1989-1998). In *Proceedings of the 10th ACM Conference on Hypertext and hypermedia*, 1999.

[7] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM Press, 2001.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[9] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000)*, 2000.

[10] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[11] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[12] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[13] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–622, 1999.

[14] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber communities. In *Proceedings of The 8th International World Wide Web Conference*, 1999.

[15] L. Lovasz and M. Plummer. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest, 1986.

[16] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[18] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, 1993.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.

[20] T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68, 2003.

[21] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

A Latent Mixed Membership Model for Relational Data

Edoardo Airoldi, David Blei, Eric Xing
School of Computer Science
Carnegie Mellon University
{eairoldi,blei,xing}@cs.cmu.edu

Stephen Fienberg
Department of Statistics, and
School of Computer Science
Carnegie Mellon University
fienberg@stat.cmu.edu

ABSTRACT

Modeling relational data is an important problem for modern data analysis and machine learning. In this paper we propose a Bayesian model that uses a hierarchy of probabilistic assumptions about the way objects interact with one another in order to learn latent groups, their typical interaction patterns, and the degree of membership of objects to groups. Our model explains the data using a small set of parameters that can be reliably estimated with an efficient inference algorithm. In our approach, the set of probabilistic assumptions may be tailored to a specific application domain in order to incorporate intuitions and/or semantics of interest. We demonstrate our methods on simulated data and we successfully apply our model to a data set of protein-to-protein interactions.

Keywords

latent mixed-membership, hierarchical mixture model, variational inference, relational data, protein-protein interactions.

1. INTRODUCTION

Modeling relational data is an important problem for modern data analysis and machine learning. Many data sets contain interrelated observations. For example, scientific literature connects papers by citation, web graphs connect pages by links, and protein-protein interaction data connect proteins by physical interaction records. Such data violate the classical exchangeability assumptions made in machine learning and statistics; moreover, the relationships between data are often of interest as observations themselves. One may try to predict citations of newly written papers, predict the likely links of a web-page, or cluster proteins based on patterns of interaction between them.

There is a history of probabilistic models for relational data analysis in Statistics. Part of this literature is rooted in the stochastic block modeling ideas from psychometrics and sociology. These ideas are due primarily to Holland and Leinhardt, e.g., [12], and later elaborated upon by others, e.g., see [8, 23, 20, 11]. In machine learning, Markov random networks have been used for link prediction [21] and the traditional block models from Statistics have been extended with nonparametric Bayesian priors [13].

In this paper, we develop a mixed membership model for analyzing patters of interaction between data. Mixed membership models for soft classification have emerged as a powerful and popular analytical tool for analyzing large databases involving text [2], text and references [4, 7], text and images [1], multiple disability measures [6, 15], and genetics information [19, 18, 24]. These models use a simple generative model, such as bag-of-words or naive Bayes, embedded in a hierarchical Bayesian framework involving a latent variable structure; this induces dependencies and introduces statistical control over the estimation of what might otherwise be an extremely large set of parameters.

We propose a Bayesian model that uses a hierarchy of probabilistic assumptions about how objects interact with one another in order to learn latent groups, their typical interaction patterns, and the degree of membership of objects to groups. Given data, we find an approximate posterior distribution with an efficient variational inference algorithm. In our approach, the set of probabilistic assumptions may be tailored to a specific application domain in order to incorporate semantics of interest. We demonstrate our methods on simulated data, and we successfully apply the model to a data set of protein-protein interactions.

2. THE MODEL

In this section, we describe a probabilistic model of interaction patterns in a group of objects. Each object can exhibit several patterns that determine its relationships to the others. We will use protein-protein interaction modeling as a working example; however, the model can be used for any relational data where the primary goal of the analysis is to learn latent group interaction patterns and mixed group membership of a set of objects.

Suppose we have observed the physical interactions between N proteins [1]. We represent the interaction data by an $N \times N$ binary adjacency matrix $\boldsymbol{r}$ where $r_{i,j} = 1$ if the ith protein interacts with the jth protein. Usually, an interaction between a pair of proteins is indicative of a unique

[1]Such information can be obtained experimentally with "yeast two-hybrid" tests and others means, and in practice the data may be noisy. For simplicity, we defer explicit treatment of observation noise, although plugging in appropriate error processes is possible.

Link-KDD '05, August 21, 2005, Chicago, Illinois, USA.

biological function they both involve; it may be possible to infer the functional classes of the study proteins from the protein interactions.

In a complex biological system, many proteins are functionally versatile and can participate in multiple functions or processes at different times or under different biological conditions. Thus, when modeling functional classes of the proteins, it is natural to adopt a flexible model which allows multiple scenarios under which a protein can interact with its partners. For example, a signal transduction protein may sometimes interact with a cellular membrane protein as part of a signal receptor; at another time, it may interact with the transcription complex as an auxiliary transcription factor. By assessing the similarity of observed protein-to-protein interaction patterns, we aim to recover the latent function groups and the degree with which the proteins take part in them.

In the generative process, we model the observed adjacency matrix as a collection of Bernoulli random variables. For each pair of objects, the presence or absence of an interaction is drawn by (1) choosing a latent class for each protein from a protein-specific distribution and (2) drawing from a Bernoulli distribution with parameter associated with the pair of latent classes. A protein represents several functional groups through its distribution of latent classes; however, each protein participates in one function when determining its relationship to another.

For a model with K groups, the parameters are K-dimensional Dirichlet parameters α, a $K \times K$ matrix of Bernoulli parameters η, and $\rho \in [0, 1]$ which is described below. Each θ_i is a Dirichlet random variable (i.e., a point on the $K-1$ simplex) and each z_{ij1} and z_{ij2} are indicators into the K groups. The generative process of the observations, $r_{(N \times N)}$, is as follows:

1. For each object $i = 1, \ldots, N$:
   1.1. Sample $\boldsymbol{\theta}_i \sim \mathit{Dirichlet}\,(\boldsymbol{\alpha})$.
2. For each pair of objects $(i, j) \in [1, N] \times [1, N]$:
   2.1. Sample group $\boldsymbol{z}_{i,j,1} \sim \mathit{Multinomial}\,(\boldsymbol{\theta}_i, 1)$
   2.2. Sample group $\boldsymbol{z}_{i,j,2} \sim \mathit{Multinomial}\,(\boldsymbol{\theta}_j, 1)$
   2.3. Sample $r_{i,j} \sim \mathit{Bernoulli}\ \left(\rho\, \eta_{z_{i,j,1}, z_{i,j,2}} + (1 - \rho)\, \delta_0 \right)$

The parameter ρ controls how often a zero is due to noise and how often it occurs as a function of the constituent proteins' latent class memberships in the generative process. In turn, this leads to ones in the matrix being weighted more as ρ decreases, and allows for the model to pick up sparsely interconnected clusters. For the rest, the model uses three sets of latent variables. The $\boldsymbol{\theta}_i$s are sampled once for the entire collection of observations; the $\boldsymbol{z}_{i,j,1}$s and $\boldsymbol{z}_{i,j,2}$s are sampled once for each protein-protein interaction variable $r_{i,j}$.

The generative process described above leads to a joint probability distribution over the observations and the latent variables,

$$p(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{z}_1, \boldsymbol{z}_2 | \boldsymbol{\alpha}, \boldsymbol{\eta}) = \prod_{i=1}^{N} p(\boldsymbol{\theta}_i|\boldsymbol{\alpha}) \prod_{j=1}^{N} p(\boldsymbol{z}_{i,j,1}|\boldsymbol{\theta}_i) \times \times p(\boldsymbol{z}_{i,j,2}|\boldsymbol{\theta}_j)\, p(r_{i,j}|\boldsymbol{z}_{i,j}, \boldsymbol{\eta}).$$

The marginal probability of the observations is not tractable to compute,

$$p(\boldsymbol{r}|\boldsymbol{\alpha}, \boldsymbol{\eta}) = \int_{\Theta} \int_{Z} \prod_{i=1}^{N} p(\boldsymbol{\theta}_i|\boldsymbol{\alpha}) \prod_{j=1}^{N} p(\boldsymbol{z}_{i,j,1}|\boldsymbol{\theta}_i) \times \times p(\boldsymbol{z}_{i,j,2}|\boldsymbol{\theta}_j)\, p(r_{i,j}|\boldsymbol{z}_{i,j}, \boldsymbol{\eta})\, d\boldsymbol{z}\, d\boldsymbol{\theta}.$$

We carry out *approximate* inference and parameter estimation to deal with this issue.

The only input to this model is the number of groups. The goal is to learn the posterior distribution of the membership proportions of each protein and the group interaction probabilities. We will focus on the interpretability of these quantities, e.g., consistent functional annotations of the proteins within groups. Note that there are several ways to select the number of groups. For example, [13] uses a nonparametric Bayesian prior for a single-membership block model.

In our fully generative approach, it is possible to integrate outside information about the objects into the hierarchy of probabilistic assumptions. For example, we can include outside information about the proteins into the generative process that includes the linkage. In citation data, document words can be modeled along with how the documents cite each other.

3. INFERENCE AND ESTIMATION

In this section we present the elements of approximate inference essential for learning the hyper-parameters of the model and inferring the posterior distribution of the degrees of membership for each object.

In order to learn the hyper-parameters we need to be able to evaluate the likelihood, which involves a non-tractable integral as we stated above—see equation. In order to infer the degrees of membership corresponding to each object, we need to compute the posterior degrees of membership given the hyper-parameters and the observations

$$p(\boldsymbol{\theta}|\boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{r}|\boldsymbol{\alpha}, \boldsymbol{\eta})}{p(\boldsymbol{r}|\boldsymbol{\alpha}, \boldsymbol{\eta})}, \tag{1}$$

Using variational methods, we can find a lower bound of the likelihood and approximate posterior distributions for each object's membership vector.

The basic idea behind variational methods is to posit a variational distribution on the latent variables $q(\boldsymbol{\theta}, \boldsymbol{z})$, which is fit to be close to the true posterior in Kullback-Leibler (KL) divergence. This corresponds to maximizing a lower bound, $\mathbb{L}\left[\, \boldsymbol{\gamma}, \boldsymbol{\phi}\, ;\, \boldsymbol{\alpha}, \boldsymbol{\eta}\, \right]$, on the log probability of the observations given by Jensen's inequality:

$$\begin{aligned} \log p(\boldsymbol{r}|\boldsymbol{\alpha}, \boldsymbol{\eta}) \geq & \sum_{i=1}^{N} \mathbb{E}_q \left[\, \log p(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i) \,\right] + \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{E}_q \left[\, \log p(\boldsymbol{z}_{i,j,1}|\boldsymbol{\theta}_i) \,\right] + \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{E}_q \left[\, \log p(\boldsymbol{r}_{i,j}|\boldsymbol{z}_{i,j}, \boldsymbol{\eta}) \,\right] + \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{E}_q \left[\, \log p(\boldsymbol{z}_{i,j,2}|\boldsymbol{\theta}_j) \,\right] - \\ & - \mathbb{E}_q \left[\, \log q(\boldsymbol{\theta}, \boldsymbol{z}) \,\right]. \end{aligned}$$

1. initialize $\boldsymbol{\gamma}_{ig}^0 = \frac{2N}{K}$ for all i, g
2. **repeat**
3. **for** $i = 1$ to N
4. **for** $j = 1$ to N
5. get **variational** $\boldsymbol{\phi}_{ij1}^{t+1}$ and $\boldsymbol{\phi}_{ij2}^{t+1} = f(r_{ij}, \boldsymbol{\gamma}_i^t, \boldsymbol{\gamma}_j^t, \boldsymbol{\eta}^t)$
6. partially update $\boldsymbol{\gamma}_i^{t+1}$, $\boldsymbol{\gamma}_j^{t+1}$ and $\boldsymbol{\eta}^{t+1}$
7. **until** convergence

1. initialize $\phi_{ij1g}^0 = \phi_{ij2h}^0 = \frac{1}{K}$ for all g, h
2. **repeat**
3. **for** $g = 1$ to K
4. update $\phi_{ij1g}^{s+1} \propto f_1(\boldsymbol{\phi}_{ij2}^s, \boldsymbol{\gamma}, \boldsymbol{\eta})$
5. normalize $\boldsymbol{\phi}_{ij1}^{s+1}$ to sum to 1
6. **for** $h = 1$ to K
7. update $\phi_{ij2h}^{s+1} \propto f_2(\boldsymbol{\phi}_{ij1}^s, \boldsymbol{\gamma}, \boldsymbol{\eta})$
8. normalize $\boldsymbol{\phi}_{ij2}^{s+1}$ to sum to 1
9. **until** convergence

Figure 1: Left: The two-layered variational inference for γ and ϕ. The inner layer consists of Step 5. The function f is described in details in the right panel. Right: Inference for the variational parameters $(\boldsymbol{\phi}_{ij1}, \boldsymbol{\phi}_{ij2})$ corresponding to the basic observation $r_{i,j}$. This is the detailed description of Step 5. in the left panel. The functions f_1 and f_2 are updates for ϕ_{ij1g} and ϕ_{ij2h} described in the text of Section 3.1.

where the expectations are taken with respect to $q(\boldsymbol{\theta}, \boldsymbol{z})$. We choose a fully factorized variational distribution such that this optimization is tractable.

3.1 Variational Inference

The fully factorized variational distribution q is as follows.

$$\begin{aligned} q(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) &= \prod_{i=1}^{N} q(\boldsymbol{\theta}_i|\boldsymbol{\gamma}_i) \prod_{j=1}^{N} q(\boldsymbol{z}_{i,j,1}|\boldsymbol{\phi}_{i,j,1})\, q(\boldsymbol{z}_{i,j,2}|\boldsymbol{\phi}_{i,j,2}) \\ &= \prod_{i=1}^{N} Dirichlet\,(\boldsymbol{\theta}_i|\boldsymbol{\gamma}_i) \;\times \\ &\quad \times \prod_{j=1}^{N} \Big(Mult\,(\boldsymbol{z}_{i,j,1}|\boldsymbol{\phi}_{i,j,1})\, Mult\,(\boldsymbol{z}_{i,j,2}|\boldsymbol{\phi}_{i,j,2}) \Big) \end{aligned}$$

The lower bound for the log likelihood $\mathbb{L}[\boldsymbol{\gamma}, \boldsymbol{\phi}\,;\, \boldsymbol{\alpha}, \boldsymbol{\eta}]$ can be maximized using exponential family arguments and coordinate ascent [22]; this leads to the following updates for the variational parameters $(\boldsymbol{\phi}_{i,j,1}, \boldsymbol{\phi}_{i,j,2})$, for each pair (i, j):

$$\begin{aligned} \phi_{i,j,1,g}^* &\propto \exp\Big\{\, \psi(\gamma_{i,g}) - \psi(\sum_{g=1}^{K} \gamma_{i,g}) \,\Big\} \times \\ &\quad \times \prod_{h=1}^{K} \eta_{g,h}^{r_{i,j}\,\phi_{i,j,2,h}} \prod_{h=1}^{K} (1 - \eta_{g,h})^{(1-r_{i,j})\,\phi_{i,j,2,h}} \\ \phi_{i,j,2,h}^* &\propto \exp\Big\{\, \psi(\gamma_{j,h}) - \psi(\sum_{h=1}^{K} \gamma_{j,h}) \,\Big\} \times \\ &\quad \times \prod_{g=1}^{K} \eta_{g,h}^{r_{i,j}\,\phi_{i,j,1,g}} \prod_{g=1}^{K} (1 - \eta_{g,h})^{(1-r_{i,j})\,\phi_{i,j,1,g}} \end{aligned}$$

for $g, h = 1, \ldots, K$, and to the following updates for the variational parameters $\boldsymbol{\gamma}_i$, for each i:

$$\gamma_{i,g}^* = \alpha_t + \sum_{j=1}^{N} \phi_{i,j,1,g} + \sum_{j=1}^{N} \phi_{j,i,2,g}.$$

The vectors $\boldsymbol{\phi}_{i,j,1}$ and $\boldsymbol{\phi}_{i,j,2}$ are normalized to sum to one. The complete algorithm to perform variational inference in the model is described in detail in Figure 1. Variational inference is carried out for fixed values of $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$, in order to maximize the lower bound for the likelihood. Then we maximize the lower bound with respect to $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$. We iterate these two steps (variational inference and maximization) until convergence. The overall procedure is a variational expectation-maximization (EM) algorithm.

3.2 Remarks

The variational inference algorithm presented in Figure 1 is not the naïve variational inference algorithm. In the naïve version of the algorithm, we initialize the variational Dirichlet parameters $\boldsymbol{\gamma}_i$ and the variational Multinomial parameters $\boldsymbol{\phi}_{ij}$ to non-informative values, then we iterate until convergence the following two steps: (i) update $\boldsymbol{\phi}_{ij1}$ and $\boldsymbol{\phi}_{ij2}$ for all pairs (i, j), and (ii) update $\boldsymbol{\gamma}_i$ for all objects i. In such algorithm, at each variational inference cycle we need to allocate $NK + 2N^2K$ numbers.

The nested variational inference algorithm trades time for space thus allowing us to deal with large graphs; at each variational cycle we need to allocate $NK+2K$ numbers. The increased running time is partially offset by the fact that the algorithm can be parallelized and leads to empirically observed faster convergence rates, as we show in Figure 3. This algorithm is also better than MCMC variations (i.e., blocked and collapsed Gibbs samplers) in terms of memory requirements and/or convergence rates.

It is also important to note that the variational Dirichlet parameters $\boldsymbol{\gamma}$ and the Bernoulli parameters $\boldsymbol{\eta}$ are closely related in this model: it is necessary to keep the $\boldsymbol{\gamma}$s across variational-EM iterations in order to better inform the M-step estimates of $\boldsymbol{\eta}$. Thus, we smooth the $\boldsymbol{\gamma}$ parameters in between EM iterations instead of resetting them to a non-informative value, $2N/K$ in our model. Using a damping parameter ϵ we obtain: $\tilde{\gamma}_{i,g} = (1 - \epsilon)\,\gamma_{i,g}^* + \epsilon\,\frac{2N}{K}$.

3.3 Parameter Estimation

Using the optimal lower bound $\mathbb{L}[\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*\,;\, \boldsymbol{\alpha}, \boldsymbol{\eta}]$ as a tractable surrogate for the likelihood we here look for (pseudo) empirical Bayes estimates for the hyper-parameters. [3]

Such maximization amounts to maximum likelihood estimation of the Dirichlet parameters α and Bernoulli parameter matrix η using expected sufficient statistics, where the expectation is taken with respect to the variational distribution. Finding the MLE of a Dirichlet requires numerical optimization. [17] For each Bernoulli parameter, the ap-

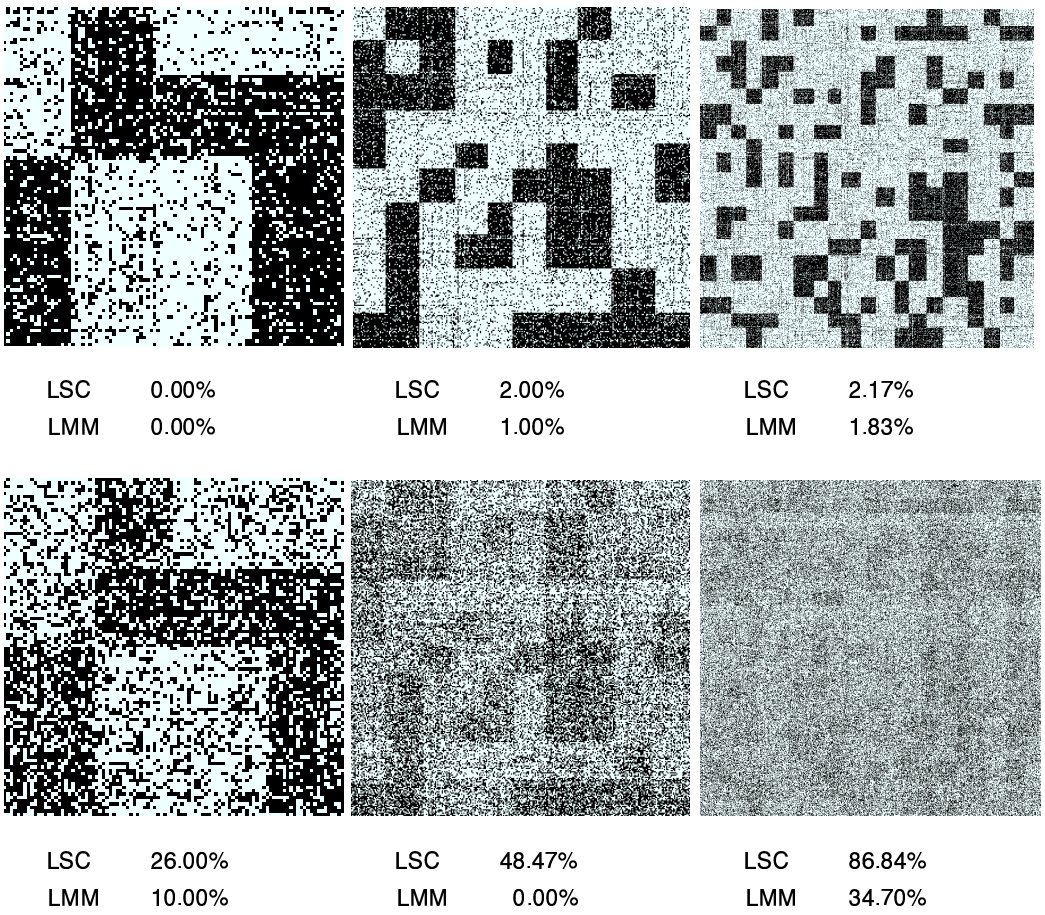


Figure 2: Error rates on simulated protein-protein interaction networks, the lower the better, for spectral clustering with local scaling (LSC) versus latent mixed-membership (LMM). From left to right: the adjacency matrices contain 100, 300 and 600 proteins and 4, 10 and 20 latent functional groups, respectively. From top to bottom: the matrices were generated using Dirichlet parameter $\alpha = 0.05$(stringent membership)$, 0.25$(more diffused membership), respectively. The proteins are re-ordered to make explicit the structure of the group interactions. The number of proteins per cluster averages 30 over all matrices. The Bernoulli probabilities in η are either 0.9 or 0.1. Random guesses about single-membership of proteins to clusters correspond to error rates of $0.75, 0.9$ and 0.95, respectively.

proximate MLE is:

$$\eta^*_{g,h} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \phi_{i,j,1,g}\, \phi_{i,j,2,h}\, r_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{N} \phi_{i,j,1,g}\, \phi_{i,j,2,h}},$$

for every index pair $(g,h) \in [1,K] \times [1,K]$.

We also smooth the probabilities of interactions between any member of group a and any member of group b, that is $\eta_{a,b}$, by assuming $\eta_{a,b} \sim Beta\,(\beta_1, \beta_2)$ for each pair of groups $(a,b) \in [1,K] \times [1,K]$. Variational inference is modified appropriately.

4. EXAMPLES AND EXPERIMENTS

We first tested our model in a controlled setting. We simulated non-contrived adjacency matrices mimicking protein-protein interactions with 100 proteins and four groups, 300 proteins and 10 groups, and 600 proteins and 20 groups. In our experiment, the signal-to-noise ratio is decreasing with the size of the problem, for a fixed Dirichlet parameter $\alpha < 1$. [2] The data are display in Figure 4, where the S/N ratio is roughly 0.5, 0.4 and 0.3 for the both the top and bottom rows, from left to right.

In Figure 4 we compare our model to spectral clustering with local scaling [25] that is particularly suited for recovering the structure of the interactions in the case when proteins take part in a single function. Note that spectral clustering (or normalized cuts) minimizes the total transition probability due to 1-step random walk of objects between clusters. Each object is assumed to have a unique cluster membership. Our model, however, is more flexible. It allows object to have different cluster membership while interacting with different objects. The simulations with the

[2]That is, a fixed $\alpha < 1$ leads to a number of active functions for each protein that increases linearly with the total number of latent functions, but the number of interactions sampled among functional groups decreases with the square of the total number of latent function, and causes an overall decrease of the informative part of the observed matrix $\boldsymbol{r}$.

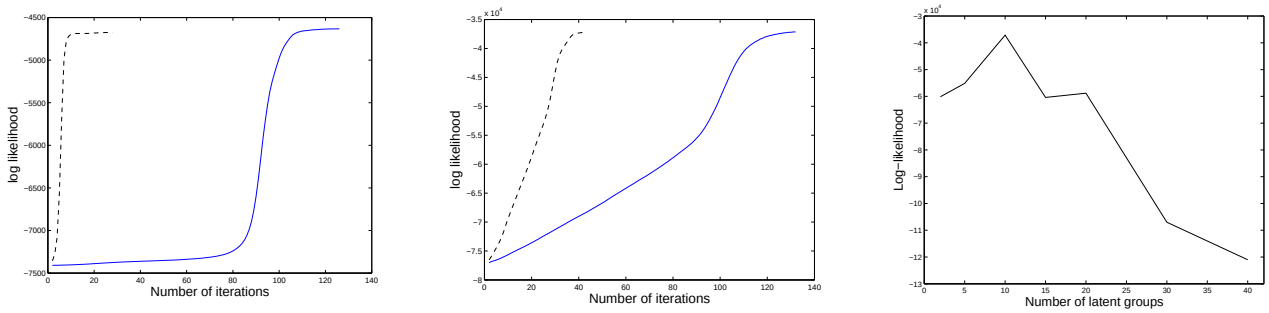


Figure 3: In the first two panels (left and center) we compare the running time of the naïve variational inference (solid line) against the running time of our enhanced (nested) variational inference algorithm (dashed line). The rightmost panel shows how the log likelihood is indicative of the latent number of functions; in the example shown the peak corresponds to the correct number of functions.

Dirichlet parameter $\alpha = 0.05$ are meant to provide mostly unique membership; spectral clustering performs well and our model has a slightly better performance. As proteins participate to more functions, that is, α increases to 0.25 in our simulations, spectral clustering is not an adequate solution anymore. Our model, on the other hand, is able to recover the mixed membership to a large degree, and performs better than spectral clustering.

In a more general formulation of our model we accommodate a collection of observations, e.g., protein-protein interaction patterns measured by different laboratories and under possible different conditions, or daily summaries of email exchanges. We used this general model to understand how the model takes advantage of the information available. Empirical results show that it is better to have a larger adjacency matrix rather than having a collection of small matrices, in order to overcome a fixed signal-to-noise ratio.

In Figure 3 compare the running time of our enhanced variational-EM algorithm to the naïve implementation. Our algorithm is more efficient in terms of space and converges faster. Further, it can be parallelized given that the updates for each interaction (i, j) are independent of one another.

4.1 Case Study: Protein-Protein Interactions

Protein-protein interactions (PPI) form the physical basis for formation of complexes and pathways which carry out different biological processes. A number of high-throughput experimental approaches have been applied to determine the set of interacting proteins on a proteome-wide scale in yeast. These include the two-hybrid (Y2H) screens and mass spectrometry methods. For example, mass spectrometry is used to identify components of protein complexes [9, 10]. High-throughput methods, though, may miss complexes that are not present under the given conditions, for example, tagging may disturb complex formation and weakly associated components may dissociate and escape detection.

The MIPS [16] database was created in 1998 based on evidence derived from a variety of experimental techniques and does not include information from high-throughput datasets. It contains about 8000 protein complex associations in yeast. We analyze a subset of this collection containing 871 proteins, the interactions amongst which were hand-curated. In Table 1 we summarize the main functions of the protein in our sub-collection, where we retained the function names in [14] where possible. Note that, since most proteins participate in more than one function, Table 1 contains more counts (2119) than proteins (871), for an average of ≈ 2.4 functions per protein. Note that the relative importance of each functional category in our sub-collection, in terms of the number of proteins involved, is different from the relative importance of the functional categories over the entire MIPS collection, as reported in [14].

Table 1: Functional Categories. In the table we report the functions proteins in the MIPS collection participate in. Most proteins participate in more than one function (≈ 2.4 on average) and, in the table, we added one count for each function each protein participates in.

#	Category	Size
1	Metabolism	125
2	Energy	56
3	Cell cycle & DNA processing	162
4	Transcription (tRNA)	258
5	Protein synthesis	220
6	Protein fate	170
7	Cellular transportation	122
8	Cell rescue, defence & virulence	6
9	Interaction w/ cell. environment	18
10	Cellular regulation	37
11	Cellular other	78
12	Control of cell organization	36
13	Sub-cellular activities	789
14	Protein regulators	1
15	Transport facilitation	41

4.1.1 Recovering the Ground Truth

Our data consists of 871 proteins participating in 255 functions. The functions are organized into a hierarchy, and the 15 functions in Table 1 are those at the top level of the hierarchy. In order to recover what we consider are the true mixed-membership vectors $\boldsymbol{\theta}_i$ corresponding to each protein, we simply normalized the number of times each protein participated in any sub-function of one of the 15 primary functions. The Dirichlet parameter α corresponding to the true mixed-membership is ≈ 0.0667. Most of the proteins in our data participate in two to four functions. In Figure 4 we show the true mixed-membership probabilities for 841 pro-

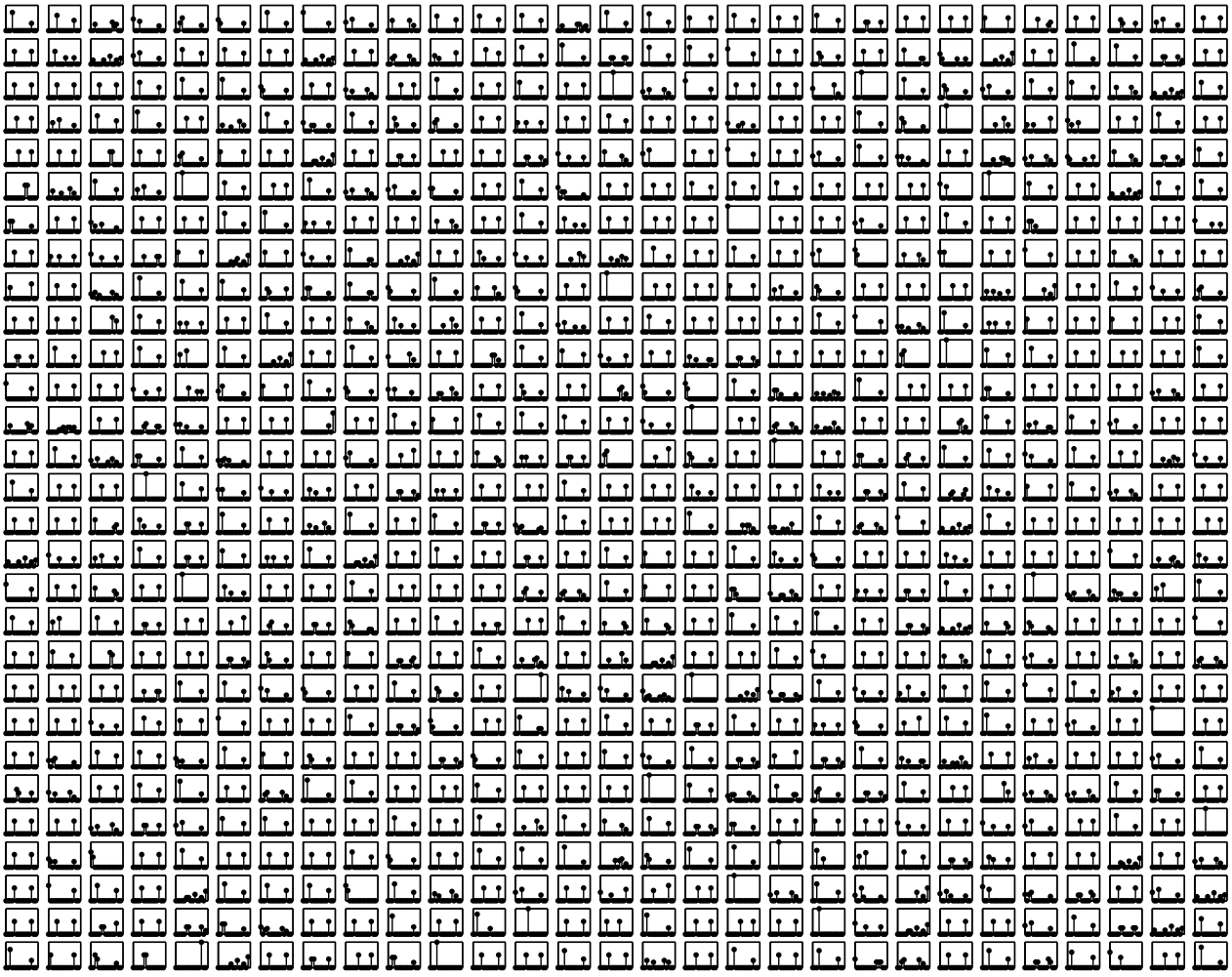


Figure 4: Mixed-membership scores estimated from hand-curated protein-protein interactions: most proteins participate in at least two functions. The figure shows 841 panels arranged in a 29 by 29 grid. Each panel plots the $\boldsymbol{\theta}_i$ of the corresponding protein in the MIPS collection. We measure probability on the Y axis and the functional group on the X axis. The functions are numbered from 1 to 15 as in Table 1.

teins.

4.1.2 Evaluating the Performance

In order to evaluate the performance of the competing methods in predicting the (possibly) multiple functional annotations of proteins we devised a very simple measure of accuracy. Briefly, we added the number of functional annotations correctly predicted for each proteins, divided by the total number of functional annotations.

Note that, given their exchangeable nature, the latent functional groups are not identifiable in our model. On the other hand, in order to compute the accuracy above we need to decide which latent cluster correspond to which functional class. We resolved the ambiguity by finding the one permutation that maximized the accuracy on the training data. We then used that permutation in order to compare predicted functional annotations to the ground truth, for all proteins.

In order to compute the accuracy of spectral clustering with local scaling, we implemented softened a soft version of it; we used the cluster predictions and the relative distances between proteins and the centroids of the clusters to obtain normalized scores (probabilities) of membership of each protein to each cluster. These mixed-membership scores enabled us to compute the accuracy measure.

4.1.3 Testing Functional Interaction Hypothesis

In order to compute the accuracy measure proposed above we need to decide which functional annotations are significantly different from zero. We used a simple statistical test to find significant functional associations: we pool all mixed-membership probabilities $\boldsymbol{\theta}_i$ together and we select the 10% most likely protein-function pairs, (i, θ_{ij}), as being significant. That is, under the assumption that most protein-function pairs are not significant, we choose the 10% most likely functional annotations to be the significant ones.

On a different note, the latent mixed-membership model is a useful tool to explore hypothesis about the nexus between latent protein interaction patterns and the functions they are able to express. For example, it is reasonable to assume

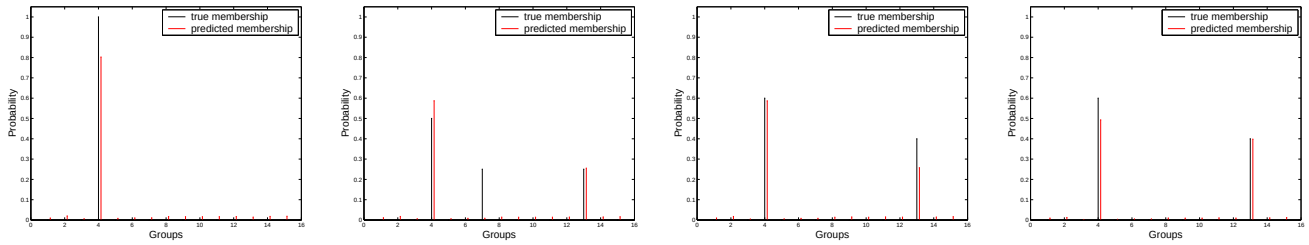


Figure 5: Predicted (red) versus true (black) mixed-membership probabilities for four example proteins.

that proteins that share a common functional annotation tend to interact with one another more often than with proteins with no functional annotations in common. In order to test this hypothesis we can fix the function interaction matrix $\boldsymbol{\eta}$ to be the identity matrix. This leads to accuracies of 43.49% for the latent mixed-membership model and of 41.67% for spectral clustering. That is, we were able to recover 504 and 483 protein-function pairs correctly out of 1159 significant, true functional annotations, for the latent mixed-membership model and (softened) spectral clustering with local scaling respectively.

4.1.4 Unsupervised Learning Experiment

In order to test the behavior of our model on real data in a situation where no information about PPI is available, we tried to recover the mixed-membership probabilities $\boldsymbol{\theta}_i$, and the function interaction matrix $\boldsymbol{\eta}$ corresponding to the hand-curated data, in a completely unsupervised fashion. We were able to recover 502 functional annotations out of 1159, which corresponds to an accuracy of 43.31%, better than spectral clustering at 41.67%.

The number of correctly identified functional annotations is comparable to the number obtained in the previous paragraph (43.49, corresponding to 504 correct annotations) under the assumption that proteins that share a common function are more likely to interact than those which do not. There is also a difference in the estimated interaction patterns between paris of latent groups, $\boldsymbol{\eta}$, which is not diagonal but rather has few positive entries arranged around two positive elements of the diagonal.

4.1.5 PPI Prediction Experiment

It is reasonable to assume that a collection of PPI may inform us on the functions protein are able to express. [5]

In order to get a feel for the prediction error associated with our model, we split the proteins into a training set and a testing set of about the same size. We then slightly modify our model in order to predict the functional mixed-membership probabilities of new proteins, i.e., those in the testing set. In particular, we use available information to learn the function interaction matrix $\boldsymbol{\eta}$, which encodes the interaction patterns between pairs of proteins as they express a corresponding pair of functions. We also consider known the functional annotations of the proteins in the training data in terms of their corresponding mixed membership probabilities $\boldsymbol{\theta}_i$. In order to estimate $\boldsymbol{\eta}$ we considered all protein pairs in the training set, and estimated the strength of the interactions between pairs of expressed functions by composing the corresponding membership probabilities of the proteins involved, under assumption of independence. In the testing phase, we fixed $\boldsymbol{\eta}$, and the $\boldsymbol{\theta}_i$ for the proteins in the training set and fit our model in order to infer the mixed- membership probability vectors of the proteins in the testing set. Alternatives are possible, where the information available is used to calibrate priors for the elements of $\boldsymbol{\eta}$, rather than fixing its values.

We were able to recover 523 functional annotations out of 1159, which for an accuracy of 45.12%. For examples of predicted mixed membership probabilities see Figure 5.

4.1.6 High-Throughput Experimental PPI

In the future we plan to explore PPI generated with high-throughput experimental methods: the tandem-affinity purification (TAP) and high-throughput mass spectrometry (HMS) complex data. [10, 9] We will use all MIPS hand-curated PPI to learn the parameters of our model, in order to provide more reliable (predicted) functional annotations for the proteins in both the TAP and HMS collections. The TAP collection contains 1363 proteins, 469 of which are contained in the MIPS hand-curated collection, whereas the HMS collection contains 1578 proteins, and shares 330 of them with the MIPS hand-curated collection.

5. DISCUSSION AND CONCLUSIONS

We have presented the latent mixed-membership model (LMM) for relational data with stochastic and heterogeneous interactions among objects. In particular, the mixed-membership assumption is very desirable for modeling real data. Given a collection of interaction patterns, our model yields posterior estimation of the multiple group membership of objects, which align closely to real world scenarios (e.g., multi-functionality of proteins). Further, our model estimates interaction probabilities between pairs of latent groups.

In simulations, our model out-performs spectral clustering both in cases when objects have single membership and in cases when objects have mixed-membership. In this latter case, the differential performance of latent mixed-membership model over spectral clustering (with local scaling) is remarkable, since spectral clustering lacks a device for capturing mixed membership. The parameter ρ of LMM enables to recover clusters whose objects are sparsely interconnected, by assigning more weight to the observed edges, i.e., the ones in the observed adjacency matrix $\boldsymbol{r}$. On the contrary, spectral clustering methods assign equal weight to both ones and zeros in the adjacency matrix $\boldsymbol{r}$, so that the classification is driven by the zeros in cases where the number of zeros is overwhelming—this may be a not desirable effect, thus it is important to be able to modulate it, e.g., with ρ.

In the case study we applied our model to the task of predicting the functional annotation of proteins by leveraging protein-protein interaction patterns. We showed how our

model provides a valuable tool to test hypothesis about the nexus between PPI and functionality. We showed how completely unsupervised inference leads to results (in terms of accuracy of the functional annotation of proteins) that are comparable to those of reasonable assumptions about how PPI leads to functionality. We also showed a way to perform cross-validation experiments in this setting, to demonstrate how it is possible to partially learn our model and make use of reliable information (about PPI) in order to infer the functionality of unlabeled proteins. Our results confirm previous findings that information about PPI alone does not lead to accurate functional annotation (in absolute terms) of unlabeled proteins. More information is needed. We plan to integrate high dimensional representation of proteins (static, non-relational) in order to boost the accuracy of functional annotation in future research.

Overall, recovering latent mixed-membership of proteins to clusters that relate to functionality provides a promising approach to learn the generative/mechanistic aspects underlying such data, which can be valuable for seeking deeper insight of the data, as well as for serving as informative priors for future learning tasks.

6. ACKNOWLEDGMENTS

The authors wish to thank Yanjun Qi, at the School of Computer Science of Carnegie Mellon University, for providing the polished version of the MIPS hand-curated data used in the PPI case study.

7. REFERENCES

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] D. Blei, A. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] B. P. Carlin and T. A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, 2005.

[4] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems 13*, 2001.

[5] M. H. Deng, K. Zhang, S. Mehta, T. Chen, and F. Z. Sun. Prediction of protein function using protein-protein interaction data. In *IEEE Computer Society Bioinformatics Conference*, 2002.

[6] E. Erosheva and S. E. Fienberg. *Classification—The Ubiquitous Challenge*, chapter Bayesian Mixed Membership Models for Soft Classification, pages 11–26. Springer-Verlag, 2005.

[7] E. A. Erosheva, S. E. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 97(22):11885–11892, 2004.

[8] S. E. Fienberg, M. M. Meyer, and S. Wasserman. Statistical analysis of multiple sociometric relations. *Journal of the American Statistical Association*, 80:51–67, 1985.

[9] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, and et. al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[10] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, and K. B. et. al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[11] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[12] P. W. Holland and S. Leinhardt. *Sociological Methodology*, chapter Local structure in social networks, pages 1–45. Jossey-Bass, 1975.

[13] C. Kemp, T. L. Griffiths, and J. B. Tenenbaum. Discovering latent classes in relational data. Technical Report AI Memo 2004-019, MIT, 2004.

[14] G. R. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the Pacific Symposium on Biocomputing*, 2004.

[15] K. G. Manton, M. A. Woodbury, and H. D. Tolley. *Statistical Applications Using Fuzzy Sets*. Wiley, 1994.

[16] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, and et. al. Mips: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32:D41–44, 2004.

[17] T. Minka. Estimating a Dirichlet distribution. Technical report, M.I.T., 2000.

[18] J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

[19] N. A. Rosenberg, J. K. Pritchard, J. L. Weber, H. M. Cann, K. K. Kidd, L. A. Zhivotovsky, and M. W. Feldman. Genetic structure of human populations. *Science*, 298:2381–2385, 2002.

[20] T. A. B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 2002.

[21] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems 15*, 2003.

[22] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, 2003.

[23] S. Wasserman and P. Pattison. Logit models and logistic regression for social networks: I. an introduction to markov graphs and p^*. *Psychometrika*, 61:401–425, 1996.

[24] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russel. Distance metric learning with applications to clustering with side information. In *Advances in Neural Information Processing Systems, 15*, 2003.

[25] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, 2004.

Discovering Missing Links in Wikipedia

Sisay Fissaha Adafre Maarten de Rijke
Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
sfissaha,mdr@science.uva.nl

ABSTRACT

In this paper we address the problem of discovering missing hypertext links in Wikipedia. The method we propose consists of two steps: first, we compute a cluster of highly similar pages around a given page, and then we identify candidate links from those similar pages that might be missing on the given page. The main innovation is in the algorithm that we use for identifying similar pages, LTRank, which ranks pages using co-citation and page title information. Both LTRank and the link discovery method are manually evaluated and show acceptable results, especially given the simplicity of the methods and conservativeness of the evaluation criteria.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*systems issues*

General Terms

Algorithms, Experimentation

Keywords

Link analysis, Wikipedia, co-citation

1. INTRODUCTION

Wikipedia, the free on-line encyclopedia, is a hypertext document with a rich link structure [17]. Though its size is very small compared to the web, its link structure shares several properties with the web. For example, some Wikipedia pages, such as pages for countries like the USA or events like World War II, are cited more often than others resulting in a skewed distribution for the incoming and outgoing links, which is a typical characteristic of the web. On the web, the motivation for creating a hyperlink tends to vary, ranging from elaboration to referential to navigational to search engine optimization. Unlike the web, most hyperlinks in Wikipedia have a more consistent and semantically meaningful interpretation and purpose. For example, in Wikipedia hyperlinks to pages for the USA or World War II are created not because of their mere popularity but rather because of their close semantic relation with the page from which they link. While some of the links in Wikipedia are navigational, the majority is conceptual rather than navigational, often providing hierarchical information (showing parent-child relationships) or pointing to more detailed descriptions or definitions of the concept denoted by the anchor text, which form integral part of the content of the page. As a result, Wikipedia may be viewed as a semantic network-like structure, where the anchor texts denote the concepts and the links represent conceptual links.

Except for general formatting guidelines and a checklist [16], there are no strict rules for editing the content of Wikipedia. Authors can freely change the content of a Wikipedia page, and adding an outgoing link to a Wikipedia page is next to trivial: by including a term in double square brackets ([[`term`]]) one creates a link to a page with `term` as its ID or title. Group consensus is the main decision making process in determining the content and outgoing link structure of a Wikipedia page. While Wikipedia's lack of strict editorial guidelines leads to surprisingly few noisy links (as far as we are able to tell from anecdotal evidence), we do see evidence for another type of problem: *missing links*. By this we mean that valuable hypertext links are missing, or that hypertext links that are present do not have the best possible anchor text (and therefore target). E.g., one would expect that

- "Kenneth Arrow . . . is an [[American]] [[economist]] . . . ",
- "Lawrence Henry Summers . . . is an American [[economist]] . . . ",
- "Gary Stanley Becker . . . is an [[American]] economist . . . ",

which refer to three American economists would all have the same link structure as the first one, i.e., "X . . . is an [[American]] [[economist]] . . . ". Consistency at this level in the link structure is obviously important for readers, but it also matters for various Wikipedia-related lexical and information extraction tasks [1]. It may also improve results of other link-based analysis techniques.

How bad is the missing links problem in Wikipedia? Studies have shown that there are significant differences in the

LinkKDD-2005, August 21, 2005, Chicago, IL, USA.

links manually assigned by different people in hypertext documents [7]. We selected a sample of 44 professional tennis players' pages from Wikipedia and examined their link structures. Though the tennis players are expected to exhibit idiosyncratic link structure depending on their country of origin, achievements, etc., we expect that they all link to the concept "Tennis", given the fact that the word "tennis" is mentioned early in their respective Wikipedia pages, mostly in the first sentence. In Wikipedia, the first paragraph usually provides a very concise description of the entity, sometimes serving as definitions. However, only 32 out of the selected 44 are linked to the Wikipedia page for "Tennis". Similarly, out of 65 randomly selected singers, only 34 are linked to the concept "singer".

Our focus in this paper is on discovering such missing links. One key challenge is the semantic ambiguity of words or phrases, which implies that not all occurrences of a word or a phrase refer to the same concept defined in Wikipedia. Furthermore, not all links have equal importance or weight. Although a term appearing in a particular page refers to a concept defined in Wikipedia, it may have little or no conceptual relation with concepts being discussed in the page. The approach adopted in this paper makes use of a clustering technique to identify related entities and search for missing links in these pages. Specifically, to create clusters of similar pages for a given page d, we use a two-step ranking mechanism, LTRank, that exploits co-citation information as well as anchor text. One of the main requirements on LTRank is that it should find pages that are not just similar to the given page d, but that if d is about a certain type of entity, then the similar pages should be about entities of the same type. E.g., if we attempt to discover links that might be missing from a page about a certain tennis player then LTRank should ideally only label pages of other tennis players as being similar.

Given a page d, and the pages most similar to d according to LTRank, we extract suggestions for links missing from d from the similar pages; the final step, then, is to identify links that are actually missing from d.

The remainder of the paper is organized as follows. In Section 2 we describe relevant features of our corpus, Wikipedia. Section 3 contains an overview of our approach to discovering missing links in Wikipedia, and in Sections 4 and 5 we detail the two key steps in our approach: given a Wikipedia page, find similar pages, and identify missing links. Section 6 contains a qualitative evaluation of the missing links we identify, while we discuss related work in Section 7 and conclude in Section 8.

2. ABOUT WIKIPEDIA

Wikipedia is a free online encyclopedia which is administered by the non-profit Wikimedia Foundation. The aim of the project to develop free encyclopedia for different languages. It is a collaborative effort of a community of volunteers, and its content can be edited by anyone. It is attracting increasing attention amongst web users and has joined the top 100 most popular sites.[1]

[1]See http://www.alexa.com/data/details/traffic_details?y=t&url=Wikipedia.org; site accessed on June 9, 2005.

2.1 The Wikipedia Corpus

As of May 16 2005, there are versions of Wikipedia in 200 languages. For the experiments in this paper, we used the English version of Wikipedia, which is the largest and contains more than 572k articles. We used the ascii text version of Wikipedia, which is available as database dump. Each entry of the encyclopedia (a page in the online version) corresponds to a single line in the text file. Each line consists of an ID (usually the name of the entity) followed by its description. The description part contains the body of the text that describes the entity. It contains a mixture of plain text and text with html tags. References to other Wikipedia pages in the text are marked using "[[" "]]" which corresponds to a hyperlink on the online version of Wikipedia. Most of the formatting information which is not relevant for the current task has been cleaned.

A Wikipedia page typically undergoes a number of revisions (on average 19.1 revisions per article for English) until a general consensus is reached among the authors regarding its content and outgoing links.

2.2 Link Structure

Wikipedia is a hypertext document with a rich link structure. As with typical web documents, Wikipedia pages differ in their content. The bulk of the Wikipedia pages provide a relatively complete description of an entity, they are *authorities* for their entities. Others act like focused *hubs*, providing a list of entities falling in a particular categories (such as list of male movie actors).

A description of an entity usually contains links to other pages within or outside of Wikipedia. The majority of these links correspond to entities, which are related to the entity being described, and have a separate entry in Wikipedia. As mentioned in Section 1, these links are used to guide the reader to a more detailed description of the concept denoted by the anchor text. This means that the links in Wikipedia typically indicate a topical association between the pages, or rather the entities described by the pages. E.g., in describing a particular person, reference will be made to such entities as country, organization and other important entities which are related to it and have entries in Wikipedia. In general, due to the peculiar characteristics of an encyclopedia corpus, the hyperlinks found in encyclopedia text are used to exemplify those instances of hyperlinks that exist among topically related entities [8, 13].

Each Wikipedia page is identified with a unique ID. These ids are formed by concatenating the words of the titles of the Wikipedia pages which are unique for each page, e.g., the page on Vincent van Gogh has "Vincent van Gogh" as its title and "Vincent_van_Gogh" as its ID. Each page may, however, be represented by different anchor texts in a hyperlink. The anchor texts may be simple morphological variants of the title such as plural form or may represent closely related semantic concept. For example, the anchor text "Dutch" points to the page for the Netherlands. In a sense, the IDs function as the canonical form for several related concepts. Although it may be difficult to say what the merits of the feature are for our task of discovering missing links, it definitely helps in minimizing the data sparseness problem, i.e., number of incoming links.

As regards the distribution of link counts, Wikipedia shows similar characteristics to the web. For example, the distribution of the incoming and outgoing links follow a power law [15].

3. OUR APPROACH

Our method for discovering links missing from a Wikipedia page consist of two main steps:

Step 1 The first step concerns identification of topically related pages, i.e., clustering.

Step 2 The second step involves identification of missing links. We identify candidate missing links and filter them through the anchor texts.

Before we jump into the details of the link discovery method, let us briefly consider a naive alternative method: simply identify and link those words or phrases in the Wikipedia pages that have an independent entry in Wikipedia. Though this method is easy to implement, there are a number of shortcomings associated with it. First, there are a number of common English words such as the article "A" and pronoun "It" that have separate entries. We do not want to create a link for every occurrence of these words. Second, most words or phrases have multiple meanings of which only a few correspond to the meaning of the Wikipedia entry. This in turn requires a separate disambiguation module, which will complicate the task.

Clustering alleviates most of the problems associated with the naive method. First, it enables us to restrict the application of the link discovery method to a particular cluster. Since links among pages entail relatedness, we prefer to establish the new links only among topically related pages. As we will see, clustering enables us to achieve this goal with a reasonable level of accuracy. Furthermore, in topically clustered Wikipedia pages, we expect that anchor texts tend to have the same meaning and refer consistently to the Wikipedia page. Moreover, in situations where an incorrect phrase is chosen, its application will be limited to that particular cluster hence avoiding global propagation of error.

Given that our overall aim is to identify links which are missing from a given page, it may not be necessary to enumerate all similar pages to a given page. Our working assumption is that a few closely related pages may suffice to identify the most important links. Furthermore, the method can be repeated multiple times to improve the link structure.

4. STEP 1: CLUSTERING PAGES

Finding the most closely related pages for a given Wikipedia page is the first step in the identification of missing links in Wikipedia hypertext corpus. One can try to find similar pages based on content, link structure or a combination of these [4, 10]. Here, we choose link structure since the main goal of the current study is recovering important missing links, therefore it is logical to look for Wikipedia pages that are similar based on their link structures. Therefore, we compute the similarity between Wikipedia pages based on the information we get from the link structure. More specifically, given a Wikipedia page, we abstract from its contents and represent it by the citations it receives, i.e., by its incoming links. We will use co-citations to identify similar pages: *two pages are similar if they are co-cited by a third*, and we assume that the co-citation counts correlate with the strength of similarity.

Round 1

1. Given a Wikipedia page d, collect all titles of the pages that link to d. Let d_t be the resulting bag of terms; we call this the full title representation of d.
2. Use Lucene to index all full representations d_t obtained in the first step.
3. For each Wikipedia page d, feed its full title representation d_t as a query against the index built in the second step, producing a ranked list $L_d = d_{t_1}, \ldots, d_{t_n}, \ldots$, of pages represented as titles of the pages that link to them.

Round 2

4. Given a Wikipedia page d, consider the titles $t_1, \ldots, t_{10}$ of the 10 pages that form the top in the list L_d generated in the third step. Represent d as the bag of terms $d_s = \{t_1, \ldots, t_{10}\}$; we call this the short title representation of d.
5. Use Lucene to index all short title representations d_s obtained in the fourth step.
6. For each Wikipedia page d, feed its short title representation d_s as a query against the index built in step five, producing a ranked list L'. Collect the top N pages in L' whose similarity score is above some threshold α. (For the experiments in this paper, we took $N = 100$, $\alpha = 0.3$.)

Figure 1: Finding similar pages using LTRank.

One naive method of identifying similar pages for a given page using co-citation information is to simply enumerate all Wikipedia pages that share a certain number of co-citations with the given page. However, pairwise comparison of all Wikipedia pages to get the most similar pages is impractical. Instead, we apply a widely used similarity measure from information retrieval. We used our own version of the Lucene search engine, an open-source, high-performance, full-text search engine written in Java [2]; it implements both vector-space and language models. We proceed in two stages: we start by representing each Wikipedia page by the IDs (or titles) of the citing pages; the pages are then indexed by treating all IDs as terms, and then every page (thus represented) is fed to Lucene as a query, producing a ranked list of pages for every page. This turns out to generate potentially off-topic lists; to overcome this issue, we need to restrict the terms used to represent pages.

Let us make things more precise now. The method that we propose is called LTRank ("Ranking based on Links and Titles"). The main steps are summarized in Figure 1. For every Wikipedia page d we create its *full title representation* d_s by collecting all titles of pages that cite d. Index all full title representations. Next, each page (represented by its citations) is submitted to the search engine as a query

in order to retrieve pages that are similar to it based on their citations. This completes Round 1 (in Figure 1), and gives us, for each Wikipedia page, a ranked list of similar pages. In most cases, however, the top ranking pages are topically closely related to the query page. In some cases, especially if the query page has many citations, the resulting list may contain a long list of unrelated pages. For example, the Wikipedia page for "Tea" has 247 incoming links. The ranked list of related pages for "Tea" that is generated using the 247 incoming links as a query contains some closely related pages, especially at the top, e.g., Coffee, Rice, Caffeine, China, Oolong, India, Xtea, Black tea. Further down the list, however, other pages which have little to do with the concept "Tea" show up, mainly concepts and people from cryptography e.g., RC6 (Block Cipher), David Wheeler (Computer Scientist).

This suggests that we should include a filtering mechanism, which is what we do in Round 2 (in Figure 1). To provide some intuitions, assume we have two Wikipedia pages together with their corresponding set of similar pages. If the two pages are similar, we expect there to be a high overlap between their corresponding sets of similar pages. Though the scenario is a bit different, a similar idea underlies the computation of SimRank [9]. In the case of SimRank, two documents are similar if the documents that cite them are similar providing a recursive mechanism for computing similarity measure. In the case of LTRank, similarity is computed using IR techniques. Therefore, in order to implement the above idea, we come up with a more lenient representation of Wikipedia pages: for each page d, we use the titles of the top 10 ranking similar pages from the list L_d obtained in Round 1; these representations are called *short title representations*. The choice of 10, though arbitrary, is an attempt to keep the balance between a concise representation for a page on the one hand, and coverage on the other hand. The short title representations are then indexed using Lucene, after which each short title representation is submitted to the search engine. The output is a ranked list of Wikipedia pages. Finally, the pages whose retrieval status value is above a certain threshold (we use 0.3) are kept, and if the list is long, we only retain the top N (we use $N = 100$).

Examination of the LTRank's output for a sample of Wikipedia pages shows that the set of similar pages identified by LTRank are (topically) closely related to the associated Wikipedia page. The list of similar pages may contain a set of homogenous items such as a list of "Tennis Players." In other cases, the list of similar pages may contain heterogeneous pages that fall under some broad concept, e.g., "Programmers," "Programming Concepts," and "Programming Languages" may fall under the broad concept "Programming," though they may look heterogeneous. For example, the list of similar pages for the Wikipedia page "Bertrand Meyer" (a programmer) consists mainly of programming concepts, rather than a set of programmers. This is sufficient for the current task which requires mainly of topical similarity: if a term is significant in one of the Wikipedia pages describing a programming concept, then we expect that the same term will be relevant if it occurs on the page for "Bertrand Meyer", a programmer.

To try and get a clearer idea of the qualitative performance of LTRank, a sample of the output for 20 pages has been selected and manually examined. We see two obvious ways to assess the output of LTRank: one is to take into account the natural semantic category to which (the topic of) a given page belongs and to demand that all similar pages found by LTRank belong to the category; the other is to simply demand that the pages returned by LTRank are relevant. Clearly, the former is more strict than the latter. For example, for "Andre Agassi" the obvious semantic category is *tennis player*, and according to the strict evaluation, similar pages should be pages of other tennis players, while the more lenient assessment criterion would also allow other types of entities, such as tennis championships that are held in different countries, such US Open, French Open etc., as long as they are relevant.

In Table 1, *Entity* refers to the Wikipedia page for which we are assessing the similar pages returned by our algorithm, *Category* refers to the category assigned to the page which is derived from the Wikipedia category information, *C-Similar* indicates the fraction of pages that fall under the category mentioned, and *Similar* is the fraction of pages that are found to be similar (without restriction to the category).

Entity	Category	C-Similar	Similar
Andre Agassi	Tennis Player	0.70	1.00
Molar Gas Constant	Thermodynamics	0.71	1.00
Etienne Ys	Prime Minister	0.00	0.86
Nine Men's Morris	Game	1.00	-
Bertrand Meyer	Programmer	0.25	1.00
Power Kite	Kite	0.75	0.75
Shiloh	Biblical Places	0.53	1.00
Marilu Henner	Actors	1.00	-
City of Darebin	Australian City	0.88	0.94
Saa	Egyptian Mythology	0.94	0.98
Jacques Ruhlmann	Designers	0.71	0.78
Kusudama	Origami	0.60	0.80
Kirkwood, Missouri	town	0.88	0.96
West European Time	Time Zone	0.64	0.64
Drill n bass	Electronic Music	0.68	0.95
Manuel Jos de Arriaga	Politician	0.59	1.00
Third	Music	0.97	1.00
Legal Code	Ethics	0.80	1.00
Alexis Arguello	Boxer	0.69	0.94
Some Kind of Wonderful	Film	0.92	1.00

Table 1: Evaluation of 20 sample clusters identified by LTRank. The average cluster size was 28 documents, the minimum size 5 documents, and the maximum size 89 documents.

For the first entity listed in Table 1, "Andre Agassi", 70% of the pages in the similar-page list are tennis players, and the rest are different tennis tournaments. In the case of "Etienne Ys", a prime minister of the Netherlands Antilles, the similar pages list contains no prime ministers but islands in the Caribbean Sea. The situation is similar for other instances shown in the table. For the current task, however, the pages on the similar-page lists are mostly relevant, as the numbers in the column labeled *Similar* indicate.

5. STEP 2: IDENTIFYING LINKS

Once we have identified the set of pages similar to a given page using LTRank, the next step is to search for important links missing from the given page. The search for missing

links is confined to this set of similar pages: our working hypothesis is that similar pages should have similar link structure.

We proceed as follows. We refer to the page that we analyze for missing links as the *main page* and to the pages identified as similar to the main page as *related pages*. Given a main page, we repeat the following steps. We take one of its related pages, and identify all outgoing links found in the related page that are absent from the main page; we also record the anchor texts for such links. The anchor texts are then searched in the main page. If an anchor is found then a link is added.

Although the anchor texts that we extract from the related pages may have different surface realizations on the main page (due to morphological variations, etc), the method works well for most instances since the set of similar pages usually contains multiple pages, which increases the likelihood of finding different surface realizations.

In general, the method identified missing links for 144,211 Wikipedia pages though not all are genuine missing links as is shown in the evaluation section. Table 2 provides some summary statistics regarding the output of the method.

Proposed Missing Links		
Min	Max	Average per Page
1	132	4
Outgoing links		
Before	After	Overlap
27	32	0.16

Table 2: Summary statistics on outgoing links and identified missing links.

In Table 2, the upper part shows the minimum, maximum and average number of missing links (per page for which links were proposed) suggested by the system. The maximum number of missing links, i.e., 132, was proposed for the page *fascism*, which contains multiple terms in the area of politics that have entries in Wikipedia. The lower part of Table 2 shows the average number of outgoing links on these pages. *Before* refers to the average number of outgoing links without the identified missing links, and *After* refers to the average with the identified missing links. For certain pages, an existing link is identified as a missing link. *Overlap* refers to the average number of this kind of links per page. These errors are mainly due to the "Redirect" facility of Wikipedia that allows two anchor texts with different IDs to point to the same Wikipedia page as will be explained in the evaluation section.

6. QUALITATIVE EVALUATION

For the task we are addressing in this paper (discovering missing links), it is difficult to compute the typical evaluation measures, i.e., precision and recall, since we do not have an exhaustive list of all the missing links. Instead, we turn to sampling-based evaluation. We take a random sample of 100 links that are identified by the method and manually examined the links. The output of the method described in Section 5 is a list of proposed links. Each entry in the list consists of an anchor text of the new link, the Wikipedia page containing the anchor text, and a similar page which has the link with this anchor text.

DIRECT3D is part of [[Microsoft]]'s [[DirectX]] [[API]]. Direct3D is only for use in Microsoft's various [[Microsoft Windows—Windows]] [[operating systems]] ([[Windows 95]] and above) and, although in a quite different version, in the [[Xbox]]. Direct3D is used to render [[3D computer graphics—three dimensional graphics]] in applications where top performance is important, such as games. Direct3D also allows applications to run fullscreen instead of embedded in a window, though they can still run in a window if programmed for that feature. Direct3D uses [[hardware acceleration]] if it is available on the graphic board.

Direct3D is a [[3D computer graphics—3D]] API. That is, it contains many commands for 3D **rendering**, but contains few commands for rendering [[2D computer graphics—2D]] graphics. Microsoft continually updates Direct3D with the latest technology available on 3D graphics cards. Direct3D offers full **vertex** software emulation but no **pixel** software emulation for features not available in hardware.
...

Figure 2: Part of a Wikipedia page for Direct3D. Missing links suggested by our method are indicated in boldface.

In order to have a consistent evaluation of the results, we used the following criteria. The first criterion checks if the anchor text has the same meaning as, or represents well, the Wikipedia page that it points to. Once the proposed link passes this test, we check if it is actually relevant; by necessity, this is a subjective decision. We examine various aspects of the proposed link in order to reduce the amount of subjectivity in determining the relevancy of a link. As described in Section 5, the link identification process involves two Wikipedia pages, i.e. a *main page* and a *related page*, which are identified as being topically related pages. Therefore, one criterion, which is less subjective, is to check if the two pages are, indeed, closely related. Otherwise, the proposed link will be based on poor evidence, and hence will be treated as noise. For example, if the pages are about presidents of two countries and the proposed link points at the page for "President", it is highly likely that the new link is relevant.

Using the above criteria, out of the sample of 100 proposed links 68 are found to be relevant. One piece of evidence that suggests that our link discovery method has to a large extent achieved its goal, is that among the links which are labelled as noisy none has been found to be caused by ambiguous anchor texts. Figure 2, which shows a portion of the Wikipedia page for "Direct3D", exemplifies the sort of links that have been identified by the method. For this particular example, the method proposed three additional links on the basis of the evidence obtained from the following three related pages, i.e., "Graphics Card", "Scene Graph", "Vertex and Pixel Shader." These are links to "Pixel," "Vertex,"

RICHARD BURNS is a [[rally]] driver from [[England]]. He was born [[January 17]] [[1971]] at the [[Royal Berkshire Hospital]],[[Reading, England]].

He started driving in field near his house at the tender age of 8 in his fathers old [[Triumph Motor Company—Triumph]] 2000. At 11 Richard joined an under 17's car club, where he became driver of the year in 1984.

Just two years later his father arranged a trip to Jan Churchill's [[Wales—Welsh]] Forest Rally School near [[Newtown]] where Richard drove a [[Ford Escort]] for the day, from that moment on Richard knew what he wanted to do.

Richard badgered his father into letting him join the [[Craven Motor Club]] in his home **town** Reading where his talent was spotted by rally raconteur and enthusiast [[David Williams]] and where he rallied the stages of Panaround, Bagshot, Mid-Wales, Millbrook, Severn Valley, Kayel Graphics and Cambrian Rally.
...

Figure 3: Part of the Wikipedia page for Richard Burns. A missing link suggested by our method is indicated in boldface.

and "Rendering" pages. All three concepts are closely related to the theme of "Direct3D." Though it might be said that the first two may be subsumed by other links in the page such as "Pixel Shaders" and "Vertex Shaders", "Rendering" seems to require a link as it is often mentioned in the page signifying its importance.

Some of the proposed links are labelled noisy due to a lack of sufficient evidence for their being relevant in the specified context. For example, a link (with anchor text "town") from the "Richard Burns" page to "town" was proposed based on the result of comparing the link structure of "Richard Burns" with its similar page "Earley" (town). Both happen to be from the same county and share some links, which explains the similarity. However, their similarity is not sufficient enough to warrant the sort of inference we are trying to make. In case of "Earley", the link "town" is mentioned as part of its definition whereas in the case of "Richard Burns" the choice of the word "town" is rather random and hence less prominent, i.e., the same information could have been expressed using other lexical items as is shown in Figure 3. Another kind of error stems from Wikipedia's redirection facility. This facility enables one to link two anchor texts with different Wikipedia IDs to the same Wikipedia pages. This feature is typically used to redirect abbreviations to the pages of their extended versions. However, it becomes a problem when the same anchor text gets two distinct IDs. Since we are using the IDs, the occurrences will be treated as different. As a result, a new link has been incorrectly suggested although the link already exists. An example of this error is "Legislative" which is represented by two IDs, "Legislative" and "Legislatur".

7. RELATED WORK

We briefly discuss two types of related work: one concerns Wikipedia, the other concerns link analysis.

7.1 Wikipedia

Though Wikipedia is very young, it has grown to be the largest free online encyclopedia in a short-period of time. Currently it has reached a level where it can support different types of research, concerning multi-authored content creation, collaborative learning, and link structure. Ciffolilli [5] describes the type of Wikipedia's community, processes of reputation and reasons for its success. Viégas et al. [14] introduce a method for visualizing edit histories of Wikipedia pages and found some collaboration patterns. Lih [11] analyzes citations of Wikipedia articles in the press and the ratio between number of edits and unique editors. Miller [12] deals with the blurring distinction between reader and author. One example of Wikipedia related research that is directly relevant to the present paper is due to Bellomi and Bonato [3], who applied link-based analysis to compute lexical authorities. Finally, Voss [15] provides a general description of Wikipedia and a broad review of Wikipedia related research.

7.2 Link analysis

In contrast to research on Wikipedia, link analysis is a relatively mature discipline with an extensive literature on the topic. One of the applications of link-based analysis techniques is identification of topically related items. Document clustering typically makes use of similarity measures that are based on terms derived from the content of the document. With the coming of the web and hypertext documents, link based techniques are gaining popularity. One prevailing assumption is that the link between entities implies certain degree of relatedness, and the link density correlates with the degree of relatedness. On the basis of these assumptions, a number of link-based techniques have been developed and applied to solve diverse problems. One of the most widely cited applications of link-based analysis techniques is to improve search engine results [4]. Traditional IR systems rely on content-based analysis techniques in order to retrieve and rank documents. With the emergence of the Web and hypertext documents, however, the relevance of a document is not solely measured by its content only but also based on the structural context which it finds itself in.

Typically, identification of a cluster involves searching for certain graph structures. Co-citation and bibliographic coupling are the two link based similarity measures. In the case of co-citation, two pages are related if they are co-cited by a third document. The strength of the relation is usually measured by the number of co-citation counts. In the case of bibliographic coupling, on the other hand, two pages are related if both cite the same document. In this paper we used co-citation ideas in developing LTRank.

Kumar et al. [10] used the concept of co-citation and graph-based techniques to identify emerging Web communities with members that are topically related. The basic idea is to identify dense bipartite graphs of small sizes and use them to identify other members of the community. Similarity is based on simple co-citation counts.

SimRank is another structural similarity measure which is based on the 'random surfer' model [9]. The intuition un-

derlying SimRank is that two objects are similar if they appear in a similar structural context. In case of a web graph, two pages are similar if they are linked to by similar pages. This formulation entails recursive computation of similarity scores. As it is impossible to compute similarity measures for all possible pairs (large n), only node pairs within certain radius apart are considered (2, 3). As pointed out in Section 4, LTRank shares a number of intuitions with SimRank. Finally, Companion is another neighborhood-based algorithm for finding related pages. For a given page, it identifies a set of closely located neighborhoods and computes authority scores for these pages and returns the highest ranking page as the most closely related page [6].

8. CONCLUSION

In this paper we addressed the problem of discovering missing links in Wikipedia. The method we proposed consists of two steps: first identify a cluster of highly similar pages around a given page, and then identify candidate links from the similar pages that might be missing on the given page. The main innovation is in the algorithm that we used for identifying similar pages, LTRank, which ranks pages using co-citation and page title information. Both LTRank and the discovery method were evaluated and showed acceptable results, especially given the simplicity of the methods and conservativeness of the evaluation criteria. Though the methods are not perfect, they could be used as an online authoring aid by revealing a ranked list of important candidate links, and the associated Wikipedia links. To some extent, this would provide a page's author with a global view of the structure of Wikipedia while locally updating or editing a page.

As to future work, we believe that there is a particularly appealing property that we get from the use of a search engine in LTRank, and that we want to explore further, i.e., term weighting. The terms, in our case, are IDs of incoming links. The search engine we use in LTRank, Lucene, uses tf.idf term weighting. The advantage of weighted incoming links becomes clear when we look at the correlation among the number of links in a page, the size of a page, and also diversity of topics. Wikipedia pages that have several outgoing links tend to be longer (on average there is about 1 hyperlink for every 17 words), and usually deal with diverse topics. In contrast, Wikipedia pages with few outgoing links tend to be shorter and homogeneous in terms of the topics dealt with. What this suggests is that there is a higher likelihood that pages that are co-cited by a page with several outgoing links are less similar than those co-cited by the short pages or pages with few outgoing links. Lucene enables us to capture these intuitions by assigning less weight to citations coming from longer pages. A brief examination of the output of LTRank supports this hypothesis, although it needs to be tested more thoroughly.

Furthermore, though we were not able to carry out a detailed theoretical comparison of LTRank method with other related techniques due to time constraints, the use of a well established information retieval technique in finding similar pages seems to add to the effiency of computation of similarity scores since it avoids pairwise comparison of Wikipedia pages (as SimRank would require). However, this should be empirically tested.

So far, we only considered co-citation as a basis for our similarity measure. However, it may be useful to take a broader view of the neighborhoods of a page, which also includes the pages cited, and search for related pages based on the extended set. This in turn would allow one to take into account the properties of another important similarity measure, bibliographic coupling, i.e., two documents are similar if they cite the same document. As mentioned previously, there is a high degree of overlap between the content and outgoing links in the Wikipedia corpus. Therefore, searching for similar pages by making use of outgoing links also partially overlaps with searching using the content of the Wikipedia pages. In addition, since we are looking for *missing* outgoing links, it may also be natural to search for pages that are similar in terms of their outgoing link pattern.

Finally, it is natural to ask whether the techniques developed in this paper can be applied in other contexts, particularly on a corpus obtained from the web. In order to shed some light into this issue, it may be worthwhile to recall those properties of Wikipedia which may have contributed (a lot) towards the results we achieved. As mentioned in previous sections, the semantic-network like nature of the Wikipedia hyperlink structure seems to be its characteristic feature which somehow distinguishes it form the web. This property in turn results in a relatively dense network structure in which the edges indicate important semantic relationships. The fact that the network is dense to some extent entails sufficiently large sets of recurrent patterns, which is important for a method based on co-citation counts. Furthermore, as in any other encyclopedia, there seems to be some kind of redundancy in Wikipedia, as certain classes of entities tend to be covered extensively. Though the content of Wikipedia can be edited by anyone, its quality is controlled indirectly through group consensus. This in turn ensures good quality content, anchor texts, and network structure. In contrast, the web tends to be more noisy without any control on its content and structure. Though some of these problems may be compensated through the mere size of the web, it may not be sufficient to match the advantages one gets from a corpus developed in a controlled environment.

9. ACKNOWLEDGMENTS

Sisay Fissaha Adafre was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Maarten de Rijke was supported by grants from NWO, under project numbers 365-20-005, 612.069.006, 220-80-001, 612.000.106, 612.000.207, 612.066.302, 264-70-050, and 017.001.190.

10. REFERENCES

[1] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach. Using Wikipedia at the TREC QA Track. In *Proceedings TREC 2004*, 2005.

[2] Apache Lucene. A high-performance, full-featured text search engine library. URL: http://lucene.apache.org, 2005.

[3] F. Bellomi and R. Bonato. Lexical authorities in an encyclopedic corpus: a case study with wikipedia. URL: http://www.fran.it/blog/2005/01/lexical-authorities-in-encyclopedic.html, 2005. Site accessed on June 9, 2005.

[4] S. Chakrabarti. *Mining the Web*. Morgan Kaufmann, 2002.

[5] A. Ciffolilli. Phantom authority, selfselective recruitment and retention of members in virtual communities: The case of Wikipedia. *First Monday*, 8(12), 2003.

[6] J. Dean and M. R. Henzinger. Finding related pages in the world wide web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1467–1479, 1999.

[7] D. Ellis, J. Furner-Hines, and P. Willett. On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases. In *SIGIR 1994: Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 51–60, 1994.

[8] R. Ghani, S. Slattery, and Y. Yang. Hypertext categorization using hyperlink patterns and meta data. In C. Brodley and A. Danyluk, editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 178–185, 2001.

[9] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, 2002.

[10] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cybercommunities. *Computer Networks*, 31(11–16):1481–1493, 1999.

[11] A. Lih. Wikipedia as participatory journalism: Reliable sources? Metrics for evaluating collaborative media as a news resource. In *Proceedings of the 5th International Symposium on Online Journalism*, 2004.

[12] N. Miller. Wikipedia and the disappearing "Author". *ETC: A Review of General Semantics*, 62(1):37–40, 2005.

[13] U. Rao and M. Turoff. Hypertext functionality: A theoretical framework. *International Journal of Human-Computer Interaction*, 1990.

[14] F. Viégas, M. Wattenberg, and D. Kushal. Studying cooperation and conflict between authors with history flow visualization. In *Proceedings of the 2004 conference on Human factors in computing systems*, 2004.

[15] J. Voss. Measuring Wikipedia. In *Proceedings 10th International Conference of the International Society for Scientometrics and Informetrics*, 2005.

[16] Wikipedia. Manual of style. URL: http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_%28links%29, 2005.

[17] Wikipedia. The Free Encyclopedia, 2005. URL: http://www.wikipedia.org.

Bayes Net Graphs to Understand Co-authorship Networks?

Anna Goldenberg
Center for Automated Learning and Discovery
Carnegie Mellon University
Pittsburgh, PA 15213, USA
anya@cs.cmu.edu

Andrew W. Moore
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
awm@cs.cmu.edu

ABSTRACT

Improvements in data collection and the birth of online communities made it possible to obtain very large social networks (graphs). Several communities have been involved in modeling and analyzing these graphs. Usage of graphical models, such as Bayesian Networks (BN), to analyze massive data has become increasingly popular, due to their scalability and robustness to noise. In the literature BNs are primarily used for compact representation of joint distributions and to perform inference, i.e. answer queries about the data. In this work we learn Bayes Nets using the previously proposed SBNS algorithm [14]. We look at the learned networks for the purpose of analyzing the graph structure itself. We also point out a few improvements over the SBNS algorithm. The usefulness of Bayes Net structures to understand social networks is an open area. We discuss possible interpretations using a small subgraph of the Medline publications and hope to provoke some discussion and interest in further analysis.

Keywords

Bayesian Networks, Structural Learning, Massive Data, Graph Analysis, Co-authorship networks

1. INTRODUCTION

The statistical literature on modeling Social Networks assumes that there are n entities called *actors* and that there exists information about binary relations between them. Binary relations are represented as an $n \times n$ matrix Y, where Y_{ij} is 1, if actor i is somehow related to j and is 0 otherwise. For example, $Y_{ij} = 1$ if "i considers j to be a friend". The entities are usually represented as nodes and the relations as arrows between the nodes. If matrix Y is symmetric, then the relations are represented as undirected arrows. More generally Y_{ij} can be real valued and not just binary, representing the strength of the relationship between actors i and j [27]. In addition, each entity can have a set of characteristics x_i such as their demographic information. Then the n dimensional vector $X = x_1, \ldots, x_n$ is fully observed covariate data that is taken into account in the model [19].

In our work, we assume that there are observations, particularly *events* relating entities (each *paper* is an event in the co-authorship dataset). However, the true underlying structure of relations between entities is not observed. We are not claiming to find the true underlying graph connecting the entities. By probabilistically modeling dependencies from the events data we aim to learn the relations robust to noise, the dependency structure and in the future predict the entities' further actions (using inference). In other words, based on the known information about simultaneous participation of entities in observed events, we construct a probabilistic model that would describe those events.

Studies on gene expression data [12] and social networks in particular suggest that correlations of entities on a local level are very important and in fact are what make up the global network [12, 7]. The SBNS algorithm [14] used here to learn the structure of Bayes Nets precisely makes use of that idea. The scalability of SBNS is achieved by exhaustively searching over structures only on the local level for a large set of small subsets of variables. The advantage of such a structural learning algorithm is that the optimization never needs to be carried out on the global scale. So, along with being computationally practical, Bayesian Networks created by our algorithm have a very natural motivation stemming from those important domains.

In this work we turn our attention to the question - how valuable are the graph structures of the Bayes Nets themselves? The resulting learned structures look like directed social networks, but the semantics behind links are different and one needs to be careful interpreting the results. In our experiments section we show two subgraphs built for the same authors of the Medline dataset that exhibit different characteristics. We observe that by learning probabilistic models we are able to draw conclusions that were not possible by simply connecting co-authors together. In fact, we believe that probabilistic graphical models that learn dependencies between entities provide a very rich structure for analysis. This work is just the beginning of the exploration in this area.

This paper is structured as follows. First we introduce notation and concepts essential to understanding the SBNS algorithm. We then provide a shorter more intuitive description of the SBNS indicating improved heuristics where applicable. Further, we give an example of a possible interpretation of the Bayes Nets in terms of the social relations in the co-authorship publications. Finally we discuss related

LinkKDD '05 August 21, 2005, Chicago, IL, USA

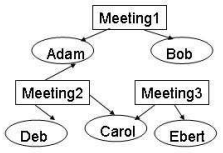
literature and conclude with our thoughts on future work.

2. BACKGROUND

In this section we introduce the terms and concepts that are most relevant to our learning algorithm. It has to be noted that the proposed algorithm readily applies only to binary, otherwise known as *transactional*, data. Thus, first we would like to introduce the general scenario where the algorithm can be applied.

2.1 Data

Assume our training data is a collection of M records of observations of N binary variables $X_1, ...X_N$. Write x_i^j as the value of X_i in the jth record where $1 \leq i \leq N$ and $1 \leq j \leq M$. Intuitively, each record denotes a collection of entities that participated in an "event". We use the words entity and *actor*, as in "social actor", interchangeably throughout the paper. The state of X_i is 1 when actor i has participated in a given event and is 0 otherwise. For example, for a citation database, if two people i and k have co-authored a paper together, then for this event (co-authorship of a given paper) their states are $X_i = 1$ and $X_k = 1$ and the states of all other variables in the database for this event are 0 ($X_t = 0, \forall t \neq i, k$). Examples of co-authorship datasets are the online library of computer science publications Citeseer, the index of online library of medical publications Medline and others, where each record is a list of co-authors of a particular paper.



	Adam	Bob	Carol	Deb	Ebert	...
Meeting1	1	1	0	0	0	0
Meeting2	1	0	1	1	0	0
Meeting3	0	0	1	0	1	0
.....	...	...	...	...	...	...

Figure 1: An example of representation (on the left) of the data (on the right). Nodes in the network are people. Rectangles are events relating them.

These datasets have one important property in common. Each record in these large datasets consists mostly of zeros: they are extremely sparse. Sparseness has been considered hazardous in statistics as it may give rise to degeneracy in models. In fact, sparseness has many advantages that are very important for computational scalability. While the problems of degeneracy arise when attempting to build a global model, sparseness is helpful to quickly identify significant local models that can later be combined into a global model. It also is instrumental in greatly improving the speed of counting that is essential in obtaining sufficient statistics.

2.2 Frequent Sets

Let the N variables be represented by integers $\{1, 2, \dots N\}$. Let the *co-occurrence frequency* of a set of attributes $S \subseteq \{1, 2, \dots M\}$ be the number of records in which all the attributes in S are simultaneously set to 1.

$$\text{freq(S)} = |\{i : \forall j \in S, x_{ij} = 1\}| \quad (1)$$

Given $s \geq 1$ we say S is a *Frequent Set of m attributes* if S contains exactly m attributes and $freq(S) \geq s$. Threshold s is called *support* in the data mining literature. Given sparse data and a support s greater than about 3, it is surprisingly easy to compute all Frequent Sets [2]. There is an abundance of literature on Frequent Sets as their collection is an essential part of the association rules algorithms [1, 2, 15] widely used in commercial data mining.

2.3 Bayes Nets

Bayesian Network (BN) is a set $\{\mathcal{G}, \theta\}$ where $\mathcal{G}$ is a Directed Acyclic Graph $\{\mathbf{V}, \mathbf{E}\}$ ($\mathbf{V}$ is a set of nodes and $\mathbf{E}$ is a set of edges) and θ is a set of parameters obtained by maximizing a Bayesian score, which is usually likelihood penalized for complexity. BNs are factored probabilistic graphical models, where the joint distribution is determined by a product of conditional probabilities, i.e.

$$P(X_1 \dots X_n) = \prod_i P(X_i | Pa(X_i)) \quad (2)$$

where $Pa(X_i) \in \mathbf{X}$ is a set of parents of the variable X_i in the DAG. Graphically, BNs are represented using directed edges from parents $Pa(X_i)$ to children X_i, for each $i = 1 \dots N$. Acyclicity of the DAG guarantees the product in Equation 2 is a coherent probability distribution. More information on Bayesian Networks can be found in [10, 17].

Note that directed arrows in the graph represent direct dependency of the outcome of variable X_i on its parents $Pa(X_i)$. The dependencies can only be described in terms of the observed data, for example in a citation database case, a relation $X_i \rightarrow X_j$, where $Pa(X_j) = X_i$, means that author X_j is likely to appear as a co-author of the paper if X_i is one of the co-authors. The dependence can also represent a negative correlation, i.e. in the above case knowing that X_i is one of the authors, would make X_j unlikely to be one of the co-authors.

3. SBNS ALGORITHM

Here we give a brief description of the SBNS (Screen-based Bayes Net Structure search) and for details we refer the reader to [14].

The SBNS algorithm is a two stage process. During the first stage, which we will call *Local Screening*, SBNS performs a Bayes Net structural search on each of the small subsets of variables defined by Frequent Sets. The resulting local structures comprise the restrictive pool of edges from which the global Bayes Net will be constructed at the second stage.

3.1 Local Screening

The intuitive idea behind the local search stage is that we do a full structural search on very small subsets of variables. One of the ways to identify the (not necessarily disjoint) subsets is to use Frequent Sets.

Screening the Frequent Sets. Suppose we have a collection of Frequent Sets $\{X : |X| = m, m \geq 2\}$. We call *screening* the process of finding the optimal Bayes Net structure for each of the Frequent Sets.

First, we screen the pairs to find pairwise correlations. We add an edge between two variables to the *Edgedump* if and only if a model that has an edge in either direction was found to have a higher score than a complete independence model between the two variables in the pair. We then in turn screen Frequent Sets of size 3, 4, etc.

It is possible that the dependencies in the Frequent Set S of size $m > 2$ are already well-explained by interactions of

order less than m. For example, suppose variables X_i, X_j and X_k co-occur frequently, but their co-occurence is well explained by the local Bayesian Network DAG structure of $X_i \leftarrow X_j \rightarrow X_k$. In that case when searching through pairs, (X_i, X_j), (X_j, X_k), (X_i, X_k), the two-way interactions will already explain all dependencies of S. In fact, only DAGs that contain a node with $m-1$ parents could be missed by not considering an m-size tuple.

We implement a *Screening* test by searching over all possible DAG structures for S and finding whether the best scoring structure has an $m-1$-parent node (we call it an *m-way* interaction). We thus allow S to pass the screening test *if and only if* S is best explained by a DAG structure containing an m-way interaction. If S passes the Screening test, all edges of the highest scoring DAG are added to the *Edgedump* – the set of edges that will eventually be considered for addition to the global Bayes Net.

3.2 Stage 2: Global Bayes Net

Once the Edgedump is created, there are several ways to construct the global Bayes Net. In this work we use the following heuristic: prioritize the edges by the score of the highest scoring m-way interaction in which they participated; create the global Bayes Net by adding the highest correlated variables first. Not all edges in the Edgedump will be added due to the acyclicity property of BNs. Note that this approach is different from the heuristic originally proposed in [14] and seemed to have resulted in higher scoring Bayes Nets in practice. We start with an empty (edgeless) global Bayesian Network and iterate through the ordered contents of the Edgedump, allowing each edge in turn to be added if and only if it improves the current score and avoids cycles. If the algorithm fails to add an edge with the direction stored in the edgedump, it tries to add the reversed edge to avoid cycles.

The proposed deterministic approach for creating a global Bayes Net is fast and performs better on average than if the edges were added randomly. However it is a simple heuristic that imposes an ordering on the variables that is not necessarily optimal.

4. NEGATIVE CORRELATION

In the previous section we pointed out that Frequent Sets allow the algorithm to consider only interactions that cause co-occurrence (and thus most likely *positive* correlations). Due to the sparse nature of the data we are not omitting the strongest correlations in general. There is, however, still a danger that if a few variables have relatively high univariate marginal probability, they could cause significant negative correlations that we would miss. Fortunately, such negative pairwise correlations can be detected cheaply by looking at a fraction of the pairs that have never occured together. We reduce the total number of entities significantly by only considering ones that occurred more than support s times in the dataset. This step is statistically justified because fewer occurrences mean lower possible mutual information. We then look at the pairs starting with the highest frequencies first.

There are two possibilities for introducing the negatively correlated pairs. One is to introduce the edges to the Edgedump from which the DAG will be constructed. Another possibility is to augment the DAG created from positive correlations. Each of the approaches has its own biases.

When we decide whether to add an edge between possibly negatively correlated variables X and Y to the Edgedump before the DAG is created, we compare the scores of the model $X-Y$ vs $X \quad Y$ and add an edge if the former scores higher (note: the direction of the edge does not matter if the scoring metric is structurally equivalent). This approach has the disadvantage of not taking into account other dependencies that may already be modeled by the existing edges in the Edgedump. It also might result in considering too many edges. However, the advantage of this approach is that when building a DAG, the pool of dependencies is more complete.

The second approach is to add edges between negatively correlated variables to the constructed DAG. In this case, we add an edge only if it does not cause a cycle and improves the score. Notice that neither of these conditions exist prior to building the DAG and are thus impossible to verify in the alternative approach describe above. The advantage of this approach is that we are likely to consider fewer pairs and thus it may be more appealing for larger networks.

5. EVALUATION CRITERION

There are several standard Bayesian scoring functions that are often used in the literature to evaluate structural learning algorithms. The structures learned were evaluated based on one of the most often used: BDEu, with an equivalent sample size of 1. The general form for the BDeu scoring function is presented in Equation 3. The BDe score was originally suggested by [8]. The u in BDeu just means a uniform prior over structures. The different scoring metrics are described in detail by [17].

$$S(G,D) = \log\left(\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\frac{1}{q_i})}{\Gamma(\frac{1}{q_i}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\frac{1}{q_i r_i}+N_{ijk})}{\Gamma(\frac{1}{q_i r_i})}\right) \quad (3)$$

where i is the ith variable, q_i - the number of states of the parents of x_i and r_i - the two states (true/false) of x_i, in our case of binary variables. Thus N_{ijk} is the number of records in our data where $X_i = k$ and $Pa(X_i)$ are in the j^{th} state.

5.1 Datasets

We have applied the Bayes Net structural learning algorithm to several co-authorship datasets (sizes are in Table 1).

Table 1: Datasets and their sizes

Datasets	Entities	Records
Institute	456	1488
NIPS	2037	1740
Medline	19499	6217
Citeseer	104801	180395

1. The *Institute Data* is a set of records of collaborations between professors and students collected from publicly available web pages listed on the Carnegie Mellon University Robotic Institute's web site.

2. The *NIPS Data Set* contains co-authorship information of the Neural Information Processing Systems con-

ference (NIPS) contained in proceedings 1-12, the pre-electronic submission era [1].

3. The *Medline Data* is a sample of the co-authorship information of the publically available medical publication database Medline.

4. The *Citeseer Data* is a set of co-publication records from the Citeseer online library and index of computer science publications. Since the entities are represented by first initial and last name, a single name might correspond to several people.

One of the key reasons why the algorithm we propose is computationally feasible is the natural tendency of large social networks to be very sparse. In other words, most of the authors tend to co-author papers with only a handful of the others considered, while very few authors co-author with a large number of others. This effect of social nets has been extensively discussed in the social networks literature [3]. Since the co-authorship data can be interpreted as a bi-partite graph, where one type of nodes are authors and the other is publications, it is interesting to note that the number of people per publication also exhibits a Power Law property: there are few publications that have many authors and majority of publications have just a few authors. In Figure 2 we provide the frequency plots for each of the datasets for both papers-per-author and authors-per-paper frequency distributions. From the plots it is apparent that co-authorship data is indeed distributed similarly to a Power Law, though some datasets tend to be particularly sparse (Medline) and some datasets tend to have heavier tails (Institute).

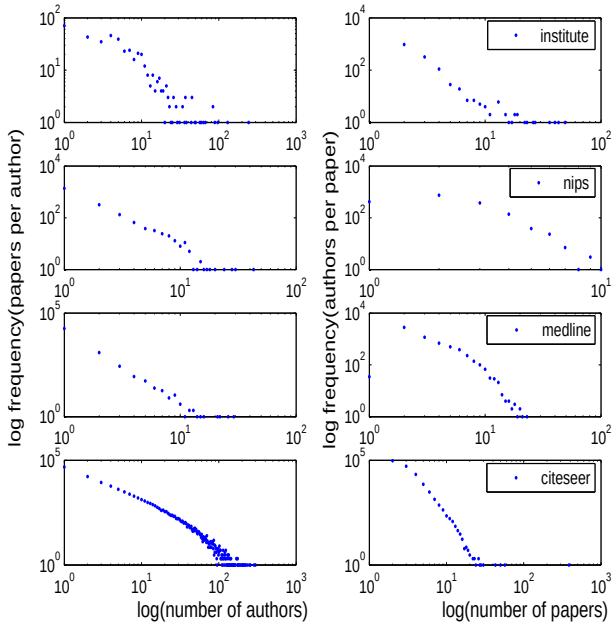


Figure 2: Marginal Frequency distribution plots

[1]This dataset was made available by Sam Roweis and can be downloaded from `http://www.cs.toronto.edu/~roweis/data.html`

5.2 Network interpretation

Usually in social science [19, 30] it is assumed that the connections are given, for example if two people have co-authored a publication they are connected by an edge in the graph. In this case the connectivity in the graph can be easily interpreted in terms of original data and the research in these fields focuses mostly on modeling the generative mechanisms and understanding the global properties of the graph. Even though the networks that we learn using the SBNS algorithm are represented as graphs connecting the same nodes, they have different semantics and one should be careful when interpreting them. We claim however that the graph structure of the learned Bayes Nets can be used effectively to gain a different view of the relations between entities (actors, people, authors).

What does the presence/absence of a link in the graph structure of the learned Bayes Net mean? First of all, the presence of a directed edge $X \rightarrow Y$ means that if the author X is known to be one of the co-authors of a paper, we can infer something about the presence of Y. By further inspection of the corresponding conditional probability table (CPT), we can say whether Y is more or less likely to be an author if X is already an author. This is a standard Bayes Net analysis. It is interesting to note, that many edges in a Bayes Net correspond to the edges in the social network, i.e. some of the edges in the social network represent significant statistical dependence between the authors. Also, due to the fact that SBNS models negative correlations as well, we can gain additional information into the set of relations that normally cannot be inferred from the social networks. For example, two doctors (from the Medline database) never co-author a paper together, but co-author quite often by themselves or with others. Knowing a few of the "negative relations" might help the network analysts to discover polarity in opinions of the corresponding doctors.

5.3 Example

To illustrate how Bayes Nets help to improve understanding about relations among doctors, we give an example of analyzing connections of a random author from the Medline publication dataset. The part of the network shown is obtained by learning the Bayes Net only on the publications that had the key word "overactive bladder", the support was set to 1 and the maximum tuple size was 3. The number of authors were consequently $16,380$ and the number of corresponding publications is $7,575$. SBNS took 1 second to learn the network. Figure 3 represents relations of the 3 levels of predecessors and successors of Alan J Wein in the learned Bayes Net.

From the part of the corresponding probability table shown in Table 2 it is evident that the presence of *Christopher R Chapple* is negatively correlated with the target *Alan J Wein* and that the presence of *Eric S Rovner* by himself is not as strong evidence for the presence of *Alan J Wein* as the presence of both *Eric S Rovner and Flavio E Trigo-Rocha*.

We also provide a social network graph where each link means co-authorship also starting with *Alan J Wein* as the main actor. We limit ourselves in this case to just people that have co-authored with Alan J Wein directly since the network grows very fast. Each link has a weight which represents how many publications the pair have appeared on as co-authors. The graph presented on Figure 5.3 appears much more interconnected with a few fully connected

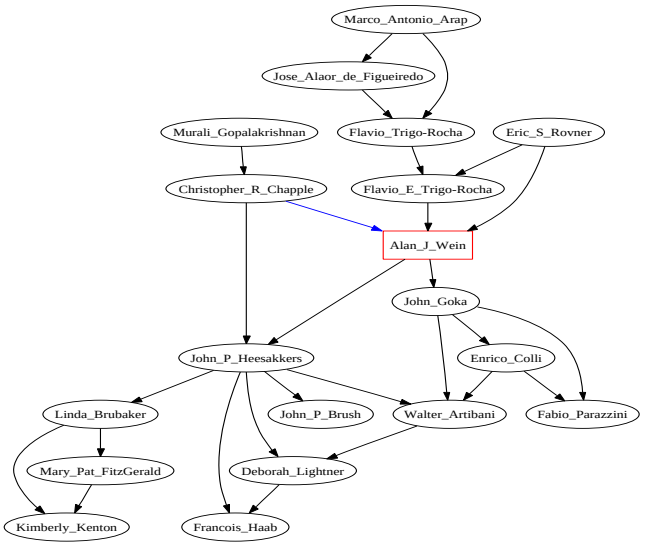


Figure 3: A part of a Bayesian Network learned from Medline publications with the keyword "overactive bladder"

			Alan J Wein	
Christopher R Chapple	Eric S Rovner	Flavio E Trigo-Rocha	0	1
0	0	0	0.997	0.003
0	1	0	0.46	0.54
0	1	1	0.33	0.67
1	0	0	0.75	0.25

Table 2: Part of a Conditional Probability Table (CPT) for *Alan J Wein* from the Bayes Net learned using SBNS

cliques. There are also several people that were not appearing in our Bayes Net. Note that the links with weights higher than 1 appear in the Bayes Net. Most links in the presented Social Network however have a weight of 1, meaning that there is not enough evidence to claim a strong dependency between co-authors. Thus, given the same data, even without increasing the support parameter, our Bayes Net learning algorithm is able to bring more clarity into the picture of relations.

5.4 Dangers in interpretation of the Bayes Net

There are certain things one should keep in mind when interpreting the Bayes Net graphs. Here we list three issues that one must be aware of, but the list might not be complete.

1. If the two nodes are not linked, it doesn't mean they are independent. It means that they are conditionally independent given their parents. Thus one must not ignore the structure of the graph when reasoning about any two nodes.

2. Proximity and number of hops in the network may not necessarily translate into the strength of a relationship as might be done in social networks. For example, in the case of the two small subgraphs presented here, from the Social Network on Figure 5.3 we see that *Deborah Lightner* has co-authored with *Alan J Wein* once and *John P Heessakkers* also co-authored with *Alan J Wein* once. In our Bayes Net on Figure 3 however *Deborah* depends on *Alan* through *John* and another parent. This does not translate into *Deborah is less likely to co-author with Alan than with John*, however it does tell us that if we know that *John* was one of the authors, knowing about *Alan* will not affect our belief in *Deborah's* presence as a co-author.

3. Our networks do not necessarily imply the causality which is usually associated with Bayes Nets. Causality needs to be tested by perturbing the evidence and seeing whether the outcome changes. We do not perform any such tests and thus in general we cannot say that the presence of X causes Y to be present, we can state however that the presence of X makes Ys presence more likely and vice versa, if that is what our conditional probability tables tell us.

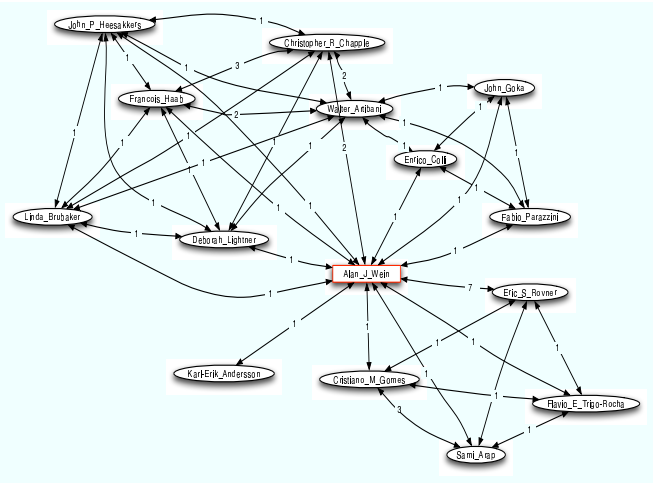


Figure 4: A part of a social network learned from Medline publications with the keyword "overactive bladder" where each link represents co-authorships and weights represent the number of co-authored publications

5.5 Global graph properties of the Bayes Nets

In terms of the global properties of the graph, we also show the graph of degree distributions for the global social network for the overactive bladder and the indegree and outdegree of the learned Bayes Nets in Figure 5.5. The Bayes Net structure seems to follow a Power Law as well. The top indegree nodes do not correspond to the top outdegree nodes. From the graph we can see that there are a few nodes with higher outdegree than the number of publications per person in the data (the social network degree distribution corresponds to precisely that). This is caused by a few of the negative correlations added, i.e. the doctors who are popular (having a high number of publications with other authors) tend to have extra edges corresponding to doctors with high number of publications whom they have never co-authored with.

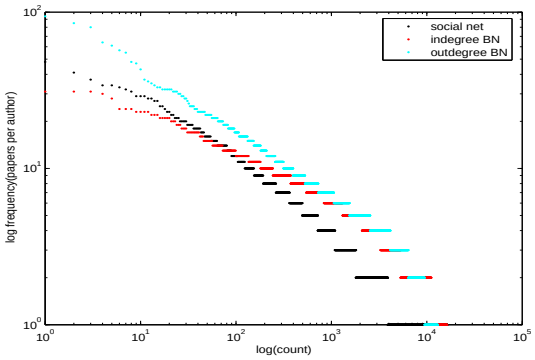


Figure 5: Degree distributions for the social network and the indegree and outdegree for the learned Bayes Net for the publications from the Medline data with the keyword "overactive bladder"

5.6 Maximum Frequent Set Size

In our experiments we tried different maximum Frequent Set sizes: ($mfss = 2 \ldots 5$). The lower bound $mfss = 2$ means that we consider only pairs of entities and thus the structure learned is based solely on two-way marginal counts. Our experiments on large datasets such as Citeseer have shown that there is an obvious loss in accuracy when high order interactions are not taken into account. Beyond a maximum Frequent Set size of 4 the number of Frequent Sets does not increase substantially in these datasets and hence the behavior of $SBNS$ changes little.

We have to note here, that there is a natural upper bound on the maximum tuple size due to the sparsity of the datasets. For example, there are $94,016$ publications in the Citeseer database that have 2 authors and only $3,022$ that have exactly 6 authors. The potential number of publications that have 6 authors, given the total number of authors in the database is 1.8×10^{27}, so the empirical number is only $(1.6 \times 10^{-22})\%$ of the total. Hence, we cannot expect a great improvement in the score of the Bayes Net when increasing the maximum tuple size, since there is not enough support for larger tuples.

5.7 Support

Lowering support greatly increases the number of Frequent Sets to be considered during screening. However, it also introduces quite a few interactions between variables that have low marginal counts. Model fitting in contingency tables in general is sensitive to very low marginal counts even if they are not zero [6]. Here we use BDeu, which is less sensitive to low counts. Despite this, it seems to be a good idea to keep support relatively large in the case of very large datasets. We have tested several support sizes on smaller datasets and found that on very sparse datasets we can use support $s = 1$ without significant overfitting. However for large datasets such as citeseer we used support $s = 3$ to reduce computational cost without affecting the BDeu score too much. We also have to note that if $s = 1$ support is used, we cannot use the approach of adding negative correlations before constructing the DAG, this approach becomes too costly. The addition of negative correlation after the construction of the DAG has showen to improve the score, while keeping the computation costs low.

5.8 Other Datasets

We have tried our algorithm on a variety of other datasets, for example IMDB (the Internet Movie DataBase) and IOBDB (the Off-Broadway shows DataBase). These datasets exhibit different properties than the publication data since it is more typical of plays and movies to have many actors. Thus, the distribution of the entities per event is different. The SBNS algorithm learns Bayes Nets that fit the data better (score higher) than the networks found by random hillclimbing. SBNS however is more time consuming since on average the data is somewhat less sparse. We are planning to do the graph analysis of these domains in the future.

6. RELATED WORK

Using Frequent Sets when learning Bayes Nets on the local scale was also explored in [25]. The goal of this work was to answer probabilistic queries on a subset of variables, thus there was no need to combine local information to obtain the joint distribution once the query size was estimated. The authors have explored Frequent Sets for quick computations of the CPTs and have noted that it is enough to look at all pairs to compute the triples without having to scan the dataset directly. The performance of Bayes Nets learned from a selection of variables was reported to be worse though close in accuracy to the inferences drawn from a Bayes Net learned on a full dataset. In [20] it has been proposed to integrate Frequent Sets as a local methodology when modelling joint distributions. This work has shown that mixture models obtained from Frequent Sets using maximum entropy are more accurate, thus supporting our claim that frequent sets contain important local information when modelling joint distributions.

One approach to speed up structural search in Bayes Nets for massive datasets has been to restrict the possible parents. The full Sparse Candidate Algorithm is presented in [13]. In its original form it is a method to speed up hillclimbing at the cost of lower performance, though in practice the performance loss was shown to be insignificant for some of the small datasets. This work is yet another motivation for us, since structural search on the local scale inadvertently restricts the number of parents. However, since on the global scale the number of parents in our Bayesian Network is not limited we perceive it as an improvement on the original Sparse Candidate algorithm.

The idea of augmenting Bayes Nets with high mutual information edges is based on the fact that such dependencies could not be accounted for in frequent sets. The fast computation used in this work is based on [22].

6.1 Statistical Network Modeling

The social network literature focuses predominantly on modeling P(Y—X), i.e. on probabilistically describing relations among actors as functions of their covariates and also properties of the graph, such as indegree and outdegree of individual nodes. A complete list of the graph-specific properties that are being modeled can be found in [30]. Thus, the models are geared to probabilistically explains the patterns of observed links and their absence between N given entities.

Several useful properties of stochastic models are listed in a brief survey work [28]. Some of them are:

- The ability to explain important properties between entities that often occur in real life such as reciprocity: if i is related to j then j is more likely to be somehow related to i; and transitivity: if i knows j and j knows k, it is likely that i knows k.
- Inference methods for handling systematic errors in the measurement of links [9]
- General approaches for parameter estimation and model comparison using Markov Chain Monte Carlo methods (e.g. [29])
- Taking into account individual variability [18] and properties (covariates) of actors [19]
- An ability to handle groups of nodes with equivalent statistical properties [31].

There are several problems with existing models such as degeneracy, analyzed by [16], and scalability, mentioned by several sources [19, 28]. The new specifications for the Exponential Random Graph Models proposed in [30] attempt to find a solution for unstable likelihoods by proposing a slightly different parametrization of the models than used previously. Experiments show that the parameters estimated using the new approach yield a smoother likelihood surface that is more robust and is less susceptible to the degeneracy problem. Scalability remains to be a major issue. Datasets with hundreds of thousands of entities are not uncommon in the Internet and co-authorship based domains. To our knowledge, there are no statistical models in the social networks literature that would scale to thousands or more actors. Parameter estimation for Markov Random Fields is well-known to be intractable in general for large number of variables due to the computational complexity of the normalization constant which requires summation over all possible graphs with N nodes. The scalability problem has also been attributed to the tendency of the models to be global, i.e. most operate on the full covariance matrices [19]. The use of MCMC approaches that tend to have slow convergence rate may also hinder computational speed of the parameter estimation in high dimensions.

One of the more recent directions is latent variable models. Those may be able to avoid the problems related to the use of Markov Random Graphs. For example, the work of [19] proposes a model in which it is assumed that each actor i has an unknown position z_i in a latent space. The links between actors in the network are then assumed to be conditionally independent given those positions. The probability of a link is a probabilistic function of the positions and actors' covariates. The latent positions are estimated from data using logistic regression. The general form of the model is:

$$logodds(y_{ij} = 1|z_i, z_j, x_{ij}, \alpha, \beta) = \alpha + \beta^T x_{ij} + d(z_i, z_j) \quad (4)$$

where $d(z_i, z_j)$ is a distance between positions of the actors in latent space. While this model is promising, it also suffers from a lack of scalability of the parameter estimation.

6.1.1 Network Modeling in Physics

The graph theoretic area of physics that studies complex systems is directly applicable to social network modeling. Though modeling of complex systems has developed seemingly in parallel to statistical modeling of social networks in social science, the findings in this area help to understand further the phenomenon of real networks organization and structure. The assumptions are the same: there are N actors (nodes) and there are M links between those nodes representing relationships among actors. The goal is also to understand and model structural properties of naturally occurring networks. The base model describing random graphs was developed by [11], where the expected number of edges in the graph is $E(N) = p(\frac{(n-1)n}{2})$, where p is the probability of having any edge, and the probability of obtaining the observed graph is $P(G_o) = p^N(1-p)^{\frac{n(n-1)}{2}-n}$. However, it was noted that the degree distribution in random graphs does not follow power law $P(k) \sim k^{-\gamma}$ common in realistic networks. Thus "scale-free networks" were introduced [4, 5]. [24] have developed a generalized random graph model where the degree distribution is given as an input parameter. Research in the field of physics gives more insight into graph growth, clusterability, graph diameter and the formation of a large component. A good summary of past and ongoing work and its relation to statistical physics is given in [3].

7. CONCLUSION

Recent work has made it computationally possible to learn Bayesian Networks from very large datasets. One of the areas where such models could be of use is social science. In particular, in this work we focus on the connection between Bayes Nets and social networks and illustrate potential interpretations of the graphical structure learned. Our simple example shows that Bayes Nets, while providing a compact representation, are a potential source for much deeper understanding of the data, such as learning about negative interactions among actors. This work is just the beginning of exploratory analysis using Bayesian Networks to model the structure of social networks themselves. We are currently collaborating with our colleagues at Pfizer to gain deeper understanding into the usefulness of this representation.

8. ACKNOWLEDGEMENTS

We would like to thank Ira Haimowitz and Pinaki Karr from Pfizer for helping with data and model interpretations. We would also like to thank Jens Nielsen and Ricardo Silva for insightful discussions.

9. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD 12*, pages 207–216, 26–28 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB 20*, pages 487–499, 12–15 1994.

[3] R. Albert and A.-L. Barabasi. Statistical mechanics of social networks. *Reviews of Modern Physics*, 74, 2002.

[4] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[5] A.-L. Barabasi, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272(1-2):173–187, 1999.

[6] Y. Bishop, S. Fienberg, and P. Holland. *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, 1977.

[7] R. Breiger. Emergent themes in social network analysis: Results, challenges, opportunities. In *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, 2003.

[8] W. Buntine. Theory refinement on Bayesian networks. In *UAI 7*, pages 52–60, 1991.

[9] C. Butts. Network inference, error, and informant (in)accuracy: a Bayesian approach. *Social Networks*, 2003.

[10] G. Cooper and E. Herskovits. A Bayesian method for constructing Bayesian belief network from databases. In *UAI 7*, pages 86–94, 1991.

[11] P. Erdos and A. Reny. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[12] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 2004.

[13] N. Friedman, I. Nachman, and D. Pe'er. Learning bayes network structure from massive datasets: The "sparse candidate" algorithm. In *UAI 15*, page 206:215, 1999.

[14] A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In 21^{st} *International Conference on Machine Learning*, 2004.

[15] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, August 2000.

[16] M. Handcock. Assessing degeneracy in statistical models of social networks. Working Paper 39, University of Washington, 2003.

[17] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian Netowrks: The combination of konwledge and statistical data. *JMLR*, 20:197–243, 1995.

[18] P. Hoff. Random effects models for network data. *Proceedings of the National Academy of Sciences*, 2003.

[19] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[20] J. Hollmen, J. Seppanen, and H. Mannila. Mixture models and frequent sets: combining global and local methods for 0-1 data. In *SIAM ICDM*, May 2003.

[21] G. Hulten and P. Domingos. Mining complex models from arbitrarily large databases in constant time. In *ACM SIGKDD 8*, pages 525–531, 2002.

[22] M. Meila. An accelerated Chow and Liu algorithm: fitting tree distributions to high dimensional sparse data. Technical Report AIM-1652, MIT, 1999.

[23] J. Moreno and H. Jennings. Statistics of social configuration. *Sociometry*, (1):342–374, 1938.

[24] M. Newman. The structure of scientific collaboration networks. In *Proceedings of the National Academy of Sciences USA*, volume 98, pages 404–409, 2001.

[25] D. Pavlov, H. Mannila, and P. Smyth. Beyond independence: probabilistic models for query approximation on binary transaction data. In *IEEE Transactions on Knowledge and Data Engineering*, September 2003.

[26] D. Pelleg and A. Moore. Using tarjan's red rule for fast dependency tree construction. In *NIPS 15*, 2002.

[27] G. Robins, P. Pattison, and S. Wasserman. Logit models and logistic regressions for social networks iii. valued relations. *Psychometrika*, 64(3):371–394, 1999.

[28] P. Smyth. Statistical modeling of graph and network data. In *IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.

[29] T. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2), 2002.

[30] T. Snijders, P. Pattison, G. Robins, and M. Handcock. New specifications for exponential random graph models. Submitted for publication, 2004.

[31] Y. Wang and G. Wong. Stochastic blockmodels for directed graphs. *Journal American Statistical Association*, 82:8–19, 1987.

Tuning Representations of Dynamic Network Data

(Extended Abstract)

Shawndra Hill
New York University
44 W 4th St., New York, NY 10012
shill@stern.nyu.edu

Deepak Agarwal Robert Bell Chris Volinsky
AT&T Labs Research
Florham Park, NJ 07934
[dagarwal,rbell,volinsky]@research.att.com

ABSTRACT

A dynamic network is a special type of network which is comprised of connected transactors which have repeated evolving interaction. Data on large dynamic networks such as telecommunications networks and the Internet are pervasive. However, representing dynamic networks in a manner that is conducive to efficient large-scale analysis is a challenge. In this paper, we represent dynamic graphs using a data structure introduced by Cortes et. al. [3]. Our work improves on their heuristic arguments by formalizing the representation with three tunable parameters. In doing this, we develop a generic framework for evaluating and tuning any dynamic graph. We show that the storage saving approximations involved in the representation do not affect predictive performance, and typically improve it. We motivate our approach using a fraud detection example from the telecommunications industry, and demonstrate that we can outperform published results on the fraud detection task.

Keywords: approximate subgraphs, dynamic graphs, graph matching, exponential averaging, fraud detection, link prediction, link analysis, statistical relational learning, transactional data streams.

1. INTRODUCTION

A graph is one way of representing complex dynamic network phenomena we encounter today. In a *dynamic graph*, nodes represent the transactors, and edges represent (directed) transactions between the transactors. A dynamic graph is built from a list of transactions with time stamps and may include other important information such as the duration of the transaction or the physical location of the transactors. Put another way, a dynamic graph is a collection of nodes and edges where the nodes and edges are subject to discrete changes, such as additions or deletions [5].

The notion of a dynamic network appears naturally in a wide range of domains. Perhaps the most obvious examples of dynamic networks are communications networks such as a telephony network or the Internet. In a telephony network data exists in the form of call detail records, which contain information on phone calls between two network transactors or IDs. Other examples of data that can be represented by dynamic graphs are author citation networks, social networks, online auctions, and disease transmission data. While data on dynamic networks are readily available, representing the dynamics in a way that is meaningful for analysis is a challenge.

One domain that may benefit from efficient dynamic graph analysis is fraud detection in telecommunications. One type of fraud is *repetitive fraud*, where we have an individual who has perpetrated some type of fraud, perhaps payment related, and has been disconnected. Sometimes the individual will attempt to set up another account, with no intention of valid payment. This individual may use methods to obscure its true identity, perhaps through identity theft, in order to obscure the fact that they are a fraudster. Therefore, we cannot use standard record-linkage techniques to link the old fraudulent account and the new account together.

However, we assume the new fraudulent ID has communication patterns similar to the old one (for example, the new ID will communicate with the same people that the old fraudulent ID did). In this way, we can use information present in the network of transactions to identify fraudsters. However, it is costly to explore all new IDs on the network in order to determine which exhibit the same network patterns as the known fraudulent ID.

Two main challenges exist for dynamic network representation. The first challenge, is to represent dynamic graphs efficiently so that the massive volumes of data in these domains can be processed in a reasonable timeframe. The second challenge, is to account for the dynamic nature of transactional data by capturing the most relevant information while eliminating spurious information that does not provide important information about the transactors.

Recently, dynamic network representations have been proposed to address these issues using node labeling schemes [6][2][7]. COI graphs [3], for example, use a general parameterized vector-based approximation that handles distributed incremental revisions to the labels. We address the aforementioned challenges with an extension of the COI method. We demonstrate our technique on a real world repetitive fraud example and show that we perform better than a representation that does not take dynamics into account. Our research makes the following contributions:

Approximation technique. We formalize the COI method, represent it as an approximation parameterized by three key

parameters, each with a clear interpretation, and provide an algorithm for how to set these parameters in a given application.

Evaluation technique. We propose an evaluation technique for parameter selection based on predictive performance on future unobserved data.

Application of technique. We apply our technique to a real world fraud detection problem and discuss its validity for a wide range of data streams. We demonstrate that predictions based on the approximated graph outperform predictions based on non-approximated data.

2. DYNAMIC GRAPH APPROXIMATION

In this research, we propose a framework for representing large dynamic networks for the class of problems where the level of analysis is at the transactor. We approximate dynamic networks by approximating individual transactors on the network using *entities*. An entity is comprised of both a specific node label (some unique identifier, also referred to as a seed node) and its corresponding local network. For example, the entity for a web user would be the user herself and all of the web pages she has visited. An entity's behavior is defined by its transactions with other nodes on the network over time, and is a subgraph of the entire communication network. The entity's level of interation with other nodes in the network evolves and may become non-existent.

Cortes, et al [3] were able capture entity change in a concise representation that changes smoothly through time. The authors use the representation to catch repetitive fraud on telecommunications networks. That paper studied the repetitive fraud problem, and approximated an entity's behavior over time by a parameterizable notion of a dynamic graph called the Communities of Interest (COI) graph.

To define a dynamic graph, we will borrow from and extend the notation of COI graphs[3]. In our approximation, the edges are represented by an exponentially weighted moving average (parameterized by θ) to summarize the activity on an edge. We prune out noise locally by defining a maximum in- or out-degree (k) for each node, with any overflow going into an aggregator node. We simplify by pruning out noise globally by removing edges which have a weight which has decayed below a threshold (ϵ), since we have high confidence that this is a stale edge. The combined parameter set $\phi = (\theta, k, \epsilon)$ enables tuning of the approximation. In our extension, we estimate the parameters to maximize our accuracy in predicting future behavior. We achieve this by maximizing a similarity score (which may be application dependent) between representations based on data in some training period and test period. The selected parameters result in a concise signature for each node and captures the most pervasive historical behavior.

We have the following objectives when building our representation:

Summarization. Represent historical behavior between two nodes in a concise manner, summarizing the relationship into a single edge with attributes.

Simplification. Prune out noise (both edges and nodes) associated with spurious transactions (such as wrong numbers) or stale relationships.

Efficiency. Handle massive data in a way that supports fast analysis and updating.

Prediction. Optimize the representation of entity behavior to maximize predictive performance of an entity's behavior in a future time period.

3. APPLYING METHODS IN CONTEXT

In this section, we will apply our method to the repetitive fraud example described in Section 1. Our goal is to identify the fraudulent individuals when they appear as a new identity by analyzing their network behavior. Our framework allows us to characterize the behavior of fraudulent individuals in a concise manner as entities, and to look for that behavior in new entities appearing on the network.

We need to show that our approximation is a good representation of an ID's behavior, in that it is a useful predictor of future behavior. In order to show this, we collected data on the usage of 1092 randomly selected active network IDS over a twelve month period. To evaluate the predictive performance of our representation, we generated three datasets using a moving window of ten consecutive months. The first nine months were used as preperiod data and the tenth consective month as test data.We apply our approximation and evaluate it using the Hellinger distance [1] as a predictive measure between approximation in the preperiod and test period, allowing us to fit our model parameter ϕ_o for this example.

3.1 Implementation

We now discuss the evaluation of our fitted parameters in our repetitive fraud application. Prior research [3] discussed implementation of COI graphs to this particular repetitive fraud problem. However, there was little discussion of the model selection, parameter fitting, properties of the approximation in general, and most importantly, the loss associated with the approximation. The prior work offers strictly heuristic arguments for using the current parameter set ϕ_c, and did not evaluate the performance of the representation other than to say that it resulted in improved fraud detection.

The current process in production uses ϕ_c and identifies 50-100 cases a day to be evaluated by fraud experts. Each case pairs a known fraudulent case with a new account that we believe might belong to the same individual. Each case is then assigned a label by an expert as to whether or not it was truly fraud. This labeling provides us with a test set to evaluate parameters, independent of the randomly selected set used to optimize the parameters.

In order to compare parameter values, we took a set of 412 actual cases identified from the current process from one week in November, 2004. Of these 412 cases, 217 of them (53%) were ultimately determined to be fraud, that is, the expert concluded thorough investigation that the new account should be shut down. For each case, we calculated the Hellinger distance between the old account and the new account for the current ϕ_c and our optimized fit ϕ_o[1]

We built a classification model using the similarity scores associated with each pair of candidate matches as attributes. We hope that our new ϕ_o results in increased score values

[1]We use the term "optimized" to refer to recommended settings based on our methodology. We realize that we do not technically optimize over three parameters, but believe our solution is optimal for typical user constraints on storage and computation.

for the cases that eventually were labelled fraud and relatively lower scores for the non-fruadulent cases. However, we realize that since the Hellinger score is monotonic with increasing k, there resulting scores are almost guaranteed to be higher for all cases (as ϕ_o has a larger k). In order to make clear that our improvement is not simply due to the monotonicity in k, we also investigate ϕ_k which has optimized values for θ and ϵ, but keeps the same k as ϕ_c.

Our goal is to see whether the new values of ϕ_k and ϕ_o allow us to discriminate fraud from non-fraud better than we are able to with ϕ_c. One way to do this is by using the Area under the ROC (AUC) curve, which measures the ability of the classifiers built using the different parameters to separate fraud and non-fraud cases. ROC curves plot the false positive rate versus the true positive rate, for different values of a score threshold. Figure 1 shows ROC curves for the three classifiers for for the model built using the Hellinger scores.

The AUC values for ϕ_o, ϕ_k, and ϕ_c are (0.831, 0.813,0.785). We find all of the ϕ are able to discriminate fraud from non-fraud. However, we find ϕ_o is the best, followed by ϕ_k and finally ϕ_c. We find that the increase in similarity scores, contributing to the higher AUC value for ϕ_o, for the optimized parameter sets over the already implemented ϕ_c are statistically significant using the paired t-test ($\alpha = 0.05$).

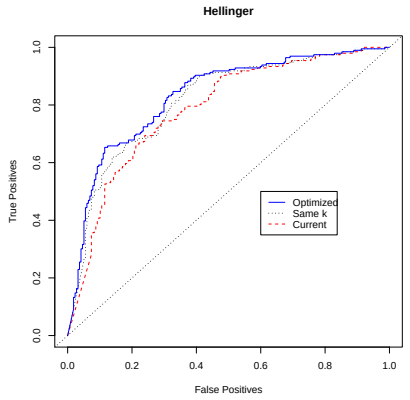


Figure 1: ***ROC curves resulting from application of new parameters to repetitive fraud example. The AUC values for ϕ_o (Optimized), ϕ_k (Same k), and ϕ_c (Current) are (0.831, 0.813,0.785).***

The implications of a high AUC and statistically significant improvement in prediction are that by setting ϕ wisely we can better rank our cases. This means that our fraud experts, who are only able to work a small number of cases per day, are better utilized. Practically, we expect a change in the parameters to result in a few extra fraud cases caught per week. In addition, the better we are at separating out fraud, the closer we get to our ultimate goal, where we have enough confidence in our scores that we can do automatic fraud detection, without a fraud expert in the loop.

4. DISCUSSION

Efficiently and effectively representing evolving dynamic networks for large scale applications is difficult. Methods such as Bayesian networks[4] and dynamic probabilistic relational models [8] consider dynamics by representing the state of the world as a set of variables, and model the probabilistic dependencies of the variables within and between time steps. While these relational representations are dynamic representations, they are more concerned with the entire network and the probabilistic relationships between the nodes on the network. When making entity approximations, we are interested in a compact representation that captures the dynamics of an individual entity on the network as opposed to the entire network.

In this study, we presented a compact dynamic network graph representation for local node analysis. We also employed the Hellinger function for assessing the correlation between local node representations of past and future node behavior on dynamic networks. We used our method to evaluate our representation on telecommunication network data.

Our main contribution is a framework for optimizing the parameter settings in a principled way for our proposed dynamic network representation. The framework can be used to evaluate any local representation that has a goal of predicting future behavior. In addition to optimizing the parameters for predictive performance as discussed above, our framework suggests the visualization of the performance gains with increasing the amount of information kept (increasing k).

5. ACKNOWLEDGEMENTS

The authors would like to thank Daryl Pregibon and Foster Provost for their very helpful comments on an early draft of this paper.

6. REFERENCES

[1] R. Beran. Minimum hellinger distance estimates for parametric models. *Annals of Statistics*, 5(3):445–463, 1977.

[2] M. A. Breuer. Coding vertexes of a graph. *IEEE Transactions on Information Theory*, IT12(2):148–153, 1966.

[3] C. Cortes, D. Pregibon, and C. Volinsky. Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics*, 12:950–970, 2003.

[4] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, Volume 5(3):142–150, 1989.

[5] D. Eppstein, Z. Galil, and G. F. Italiano. *Dynamic graph algorithms*. CRC Press, 1999.

[6] C. Gavoille and C. Paul. Distance labeling scheme and split decomposition. *Discrete Mathematics*, 273(1-3):115–130, 2003.

[7] A. Korman and D. Peleg. *Labeling Schemes for Weighted Dynamic Trees (Extended Abstract)*, volume 2719 of *Lecture Notes in Computer Science*. Springer-Verlag Heidelberg, 2003.

[8] S. Sanghai, P. Domingos, and D. S. Weld. Dynamic probabilistic relational models. In G. Gottlob and T. Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 992–1002. Morgan Kaufmann, 2003.

Email Alias Detection Using Social Network Analysis

Ralf Hölzer
Information Networking Institute
Carnegie Mellon University
Pittsburgh, PA 15213
rholzer@cmu.edu

Bradley Malin
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
malin@cs.cmu.edu

Latanya Sweeney
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
latanya@privacy.cs.cmu.edu

ABSTRACT

This research addresses the problem of correctly relating aliases that belong to the same entity. Previous approaches focused on natural language processing and structured data, whereas in this research we analyze the local association, or "social" network in which aliases reside. The network is constructed from email data mined from the Internet. Links in the network represent web pages on which two email addresses are collocated. The problem is defined as given social network S, constructed from email address collocations, and an email address E, identify any aliases for E that also appear in S. The alias detection methods are evaluated on a data set of over 14,000 University X email addresses for which ground truth relations are known. The results are reported as partial lists of k choices for possible aliases, ranked by predicted relational strength within the network. Given a source email address, a portion of all email addresses, 2%, are correctly linked to another alias that corresponds to the same entity by best rank, which is significantly better than random (0.007%) and a geodesic distance (1%) baseline prediction. Correct linkages increase to 15% and 30% within top-10 (0.07% of all emails) and top-100 rank lists (0.7% of all emails), respectively.

1. INTRODUCTION

Individuals on the Internet use aliases for various communication purposes. Aliases can be tailored to specific scenarios, which allows individuals to assume different aliases depending on the context of interaction. For example, many online users utilize aliases as pseudonyms in order to protect their true identity, such that one alias is used for web forum postings and another for e-mail correspondence. Determining when multiple aliases correspond to the same entity, or *alias detection*, is useful to a variety of both legitimate and illegitimate applications. Regardless of the intent behind alias detection, it is important to understand the extent to which the process can be automated.

When aliases are listed on the same webpage it can indicate there exists some form of relationship between them. In order to leverage this relationship, we analyze several methods for alias detection based on social network analysis [19]. Social network analysis has recently been integrated into the computer science community to model several problems, including record linkage in co-authorship networks [4] and name disambiguation [13]. We assume the network in which aliases, extracted from webpages, are situated reveal certain aspects of the social network to whom the alias corresponds.

Since many people use several email addresses for related purposes, we attempt to determine which email addresses correspond to the same entity by analyzing the relational network of addresses extracted from webpages. Email addresses, a type of alias, can be distilled from a large number of web pages, such as class rosters [18], research papers [5], resumes [12], discussion boards, or USENET message archives [7]. For this paper networks are constructed from email addresses extracted from web pages within a specific university's system. As a result, similarities in the local network surrounding each address can be exploited to determine which aliases correspond to the same entity. Furthermore, email addresses provide another useful property for determining relationships. In contrast to other identifiers, email addresses provide a unique mapping from address to a specific entity. Thus, no disambiguation is necessary when studying email addresses as identifiers for alias detection.

The remainder of this paper is organized as follows. Section 2 reviews earlier approaches to alias detection and determining importance between nodes in social networks. Novel methods based for alias detection are discussed in section 3. In addition, the graph representation of the network and the ranking algorithms are introduced. In section 4, the detection methods are evaluated on a dataset for which a large number of email aliases are known. Results and limitations of the approaches are discussed in section 5.

2. RELATED RESEARCH

Alias detection is related to the problem of alias disambiguation. The latter attempts to determine if the same alias, such as "John Smith", refers to one or multiple entities. There are certain similarities between the disambiguation and detection, and as a result, some of the methods and insights garnered from one can be applied to the other. In this section we review several approaches which have been applied to the disambiguation and detection problems. The approach of choice depends primarily on the type of underlying data to be analyzed.

Natural language processing has been successfully applied to identify whether separate writings have been authored by the same individual. Computational and statistical models were first proposed by Mosteller and Wallace [14] to solve disputes regarding the authorship of free text documents. Their models were extended by Rao et al. [17] who applied techniques from linguistics and stylometry to identify pseudonyms in a textual context on the Internet. These methods were successful in identifying aliases used by the

LinkKDD 2005, August 21, 2005, Chicago, Illinois, USA.

same individual, even if these individuals applied technical measures to disguise their identity. Novak et al. [15] have also developed text analysis algorithms to determine aliases used on the web. These algorithms are based on analyzing the text actively posted on the web under a variety of aliases.

In contrast to free text analysis, several researchers have focused on the analysis of structured bibliographies to characterize authorship. Han et al. [8] used machine learning methods to disambiguate names and pseudonyms in citation data, where an author publishes under similar but not identical names. Similarly, Pasula et al. [16] developed a probabilistic algorithm to solve the problem of ambiguous citations in scientific publications. The latter method is based on formal algorithms for linkage of similar records in databases with defined attributes set forth by Winkler [21]. Yet, these approaches are highly dependent on the structure of information surrounding the entities.

Recently, researchers have turned to social network analysis techniques for alias detection and name disambiguation [4, 3, 9, 10, 11, 13]. Similar to the previous methods, Hill [9] and Hsiung et al. have built classifiers for aliases based on relational networks that were trained in a supervised environment. For instance, Hill constructed classifiers for paper authors that are derived from co-citation data. When provided with a new paper, the author of which was unknown, the citation-based classifiers were used to determine the author. From an unsupervised perspective, Bhattacharya and Getoor [4, 3] extended Winkler's record linkage methods [21] by incorporating co-authorship link structure of the underlying data. These algorithms use an iterative process for deduplication in order to determine if two identifiers refer to the same entity. This approach is similar to alias detection, where two identifiers refer to a single real-world entity. Though this method is tailored to social networks which manifest as clique structures, alternative has been developed for name disambiguation in less centralized social systems [11, 13]. One such approach, proposed by Malin [13], is based on an importance ranking in a relational network surrounding the entity in question. The method looks at collocations and the size of the source from which identifiers are extracted. Unlike the methods proposed in this paper, these network-based approaches fail to explicitly account for the impact of source size and number of collocations independently.

Whereas the previous studies attempted alias detection and disambiguation, Adamic and Adar [1] studied methods to determine relationship importance from mailing lists and other data on the web. Their weighting scheme for predicting similarity in a social network is similar to the weighting algorithms in this paper, but only uses a single, combined measure. White and Smyth [20] have previously developed algorithms to determine importance in social networks in a more general setting. However, these algorithms for determining importance between nodes do not take any heuristics into account.

3. ALIAS DETECTION METHODS

To detect multiple aliases corresponding to the same entity via network analysis, aliases are collected from sources with collocations. In the case of email addresses the sources are web pages listing several email addresses. In this section, we describe network representation and similarity measurements between aliases pairs.

3.1 Data Representation

Let S represent the set of sources from which identifiers are extracted. Let I be the set of unique email addresses, where I_s denotes the subset of addresses listed on a source $s \in S$. The social network of email addresses is modelled as an undirected graph $G = (I, E)$. Each node $i \in I$ is a distinct email address and each edge $e_{ab} \in E$ is a list of sources in which i_a and i_b were collocated. Let $c_{ab} = |e_{ab}|$ denote the number of sources associated with each edge connecting a and b. As a corollary, the network contains an edge between each pair of email addresses that collocated on at least one. Similarly, there exists a clique (i.e. non-null edge) between all addresses on $s \in S$.

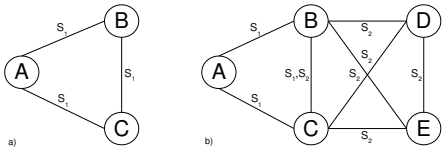


Figure 1: *a)* shows the graph with one source S_1 containing the identifiers $\{A, B, C\}$. *b)* shows the graph after a source S_2 with identifiers $\{B, C, D, E\}$ has been added.

Figure 1a) shows a network containing a single source $S_1 = \{A, B, C\}$, were A, B, and C are identifiers on the source. In Figure 1b), a second source $S_2 = \{B, C, D, E\}$ with the corresponding identifiers has been added. The identifiers B and C are collocated on two different sources, represented by the two sources listed on the edge connecting them.

3.2 Ranking Algorithms

This section describes the ranking methods. A ranking is a top-k list of possible aliases, with the most likely alias candidates at the top of the list. A shortest path algorithm is used to generate a ranking of nodes closest to a given originating node. Nodes closer to the source are favored over nodes at a further geodesic distance.

A useful measure to describe the distance between two nodes in the network is geodesic distance [19]. The geodesic distance between originating node a and destination node b is the length of the shortest path from a to b. The first approach ranks possible aliases for a source alias by geodesic distance in an unweighted network.

The goal of the subsequent ranking methods in this section is to adjust the weights on the edges connecting two nodes according the source data. Specifically, relationship strength is augmented using two heuristics: 1) the number of aliases on a source and 2) the number of collocations of aliases. For these weighting heuristics, the edges have a default weight of 1. Each method reduces the weight to a minimum of $\frac{1}{2}$. All weights are normalized and constrained to the interval $\left[\frac{1}{2}, 1\right]$. For ranking purposes, to preserve a minimum distance on each path, values lower than $\frac{1}{2}$ are not permitted.

3.2.1 Geodesic

Potential aliases are ranked from lowest to highest geodesic distance.

3.2.2 Multiple Collocation

This weighting schema models the assumption that two aliases which collocate on more than one webpage signifies a stronger relationship. For this method, the weights on the graph are calculated as

$$multicol_{ij} = \frac{1 + \prod_{i=1}^{c_{ij}} \frac{1}{2^{i-1}}}{2} = \frac{1}{2} + \frac{1}{2^{c_{ij}}}.$$

The weight is reduced according to an exponential decay function on the number of sources in common. The first source will have a large impact, whereas each additional source will have decreasing impact on the reduction of the weight. This ensures that any additional collocation will be taken into account, but with decreasing impact. As a result, the weight has an upper bound of $c_{ij} = 1$ and has an asymptotic lower bound of $\frac{1}{2}$ with each additional edge, where smaller values represent higher importance.

3.2.3 Source Size

For this weighting schema, we model the belief that the strength between two aliases is inversely correlated with the number of aliases in a source. To evaluate this assumption, edges in the graph are reweighted by setting the weight to

$$sourcesize_{ij} = 1 - \frac{1}{|s_{ij}|}.$$

Since the smallest number of identifiers on a source is 2, the minimum distance for this weight is $\frac{1}{2}$. For large values of $|s_{ij}|$, the weight is asymptotic to 1.

3.2.4 Combined

This approach integrates both of the previous assumptions. Basically, all edges between nodes i and j are weighted using minimum akin to control theory method. The weight is calculated as:

$$combined_{ij} = Max\left(1 - \sum_{i=1}^{c_{ij}} \frac{1}{\alpha \times |s_{ij}|}, \frac{1}{2}\right)$$

Each additional edge reduces the weight by a certain amount dependent on the number of identifiers on the source, such that large source sizes reduce the weight less than smaller source sizes. The resulting weight is on the interval $[\frac{1}{2}, 1]$, where, again, a smaller weight indicates higher importance. Since the weight is reduced for each additional edge and the number of edges is theoretically unlimited, the maximum reduction is upper bounded by $\frac{1}{2}$. For this research, we set $\alpha = 10$, so it takes a maximum of 20 edges with a source size of 2 into account, after which there will be no additional weight reduction.[1]

4. EXPERIMENTAL EVALUATION

This section analyzes the methods described above using email address data derived from Carnegie Mellon University (CMU) web pages. For this analysis, a dataset of CMU-specific email addresses were extracted. This dataset contains 1978 distinct email aliases, with ground truth relations known for all, which makes the dataset amenable to evaluation.

4.1 Data Set Statistics

Due to the way in which email addresses are assigned and can be chosen at CMU, each individual is assigned a unique id in the university-wide andrew.cmu.edu, or *Andrew*, email domain. Many departments have self-maintained email subdomains, which provide additional email addresses for each individual in that department. For example, a graduate student in the Department of Electrical and Computer Engineering (ECE) may use a second email address in the ece.cmu.edu subdomain in addition to Andrew. Since usernames in most of these subdomains correspond to those assigned in the Andrew system, it is possible to generate an accurate list of email aliases from the data set. All addresses that were clearly not a person, such as *root*, *webmaster*, or *cs-students* were removed from consideration. The set of aliases is summarized in table 1.

We found 18%, 45%, and 11% coverage of emails in the database for Andrew, SCS, and ECE email address, respectively. However, the percentages are a rough coverage estimate since some inactive emails no longer in the directory can exist on collected webpages. Since most people in ECE and SCS have an email address in more than one CMU subdomain and therefore have an alias in the dataset, the probability of finding at least one email address for a person in the database is higher than the percentages shown.

Total # of aliases	1978
# of distinct individuals	897
individuals with 2 aliases	767
individuals with 3 aliases	100
individuals with 4 aliases	17
individuals with 5 aliases	6
individuals with 6 aliases	6

Table 1: Aliases in the Carnegie Mellon dataset.

In order to determine if collocated email addresses on webpages provide a foundation for non-random networks, several simple tests were run. To see whether an average individual at Carnegie Mellon can be found on the web (and therefore in the graph), several email directories where compared to the contents in the database. These directories included a full list of all active Andrew email accounts in the university-wide email system, all email accounts in the Department of Computer and Electrical Engineering, and the emails listed in the directory of the School of Computer Science (SCS). Table 2 shows the percentage of emails in these directories that are contained in the graph.

Directory	In DB	# of emails	Percentage
Andrew	38764	6835	18%
SCS	903	2003	45%
ECE	161	1504	11%

Table 2: Percentage of email addresses in database, per directory.

Table 3 shows the path lengths generated by running all-pairs shortest paths on all email addresses in the data set. Intuitively, the average path length between any two email addresses across the entire data set should be higher than the average path length between email addresses in a certain department. The results in Table 3 support this hypothesis. The networks where generated by selecting only those paths that have both a source and the destination address with the corresponding subdomain. The intermediary nodes did not need to be in that specific subdomain.

4.2 Results

The ranking methods described above were then applied to the dataset for evaluation. Several different statistics are presented in this section that support the source size and multiple collocation heuristics.

4.2.1 Geodesic Alias Distances

[1] In the data set analyzed, the maximum number of collocations for two aliases was less than 20 pages. Thus, value of 10 should be adjusted for different data sets where large number of collocations are observed.

Subdomain	Avg.	Max.	Stddev.	emails
All	4.15	12	1.18	14766
cs.cmu.edu	3.76	11	1.20	2897
ece.cmu.edu	3.14	8	1.25	514
cald.cs.cmu.edu	1.70	2	0.63	11
privacy.cs.cmu.edu	2.63	4	0.71	99
speech.cs.cmu.edu	1.82	6	1.60	42

Table 3: Path lengths per subdomain.

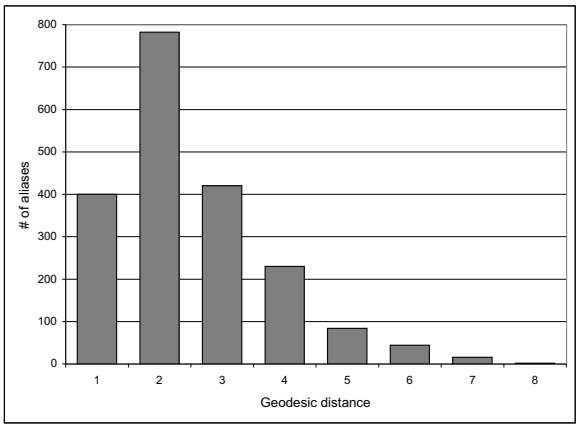


Figure 2: Geodesic distance of all email address pairs.

One hypothesis made earlier was that aliases corresponding to the same entity are close to each other within the network. Figure 2 shows the distances between all pairs of email addresses in the data set. More than 50% of email pairs corresponding to the same entity are within a geodesic distance of two. Moreover, about 400 aliases are within a distance of one. On average, pairs of an entity's email addressed were 2.5 geodesic distance from each other, which is significantly shorter than an average of 4.15 for randomly chosen email pairs. This indicates that pairs of email aliases which correspond to the same entity occur within relatively close proximity of each other.

4.2.2 Method Comparison

Figure 3 summarizes the ranking results. For each known email address in the set of well known aliases, several top-k rankings were generated. These rankings consist of k possible alias candidates, as determined by each method described in Section 3. The top-k lists contain the likely aliases ordered by importance in descending order. The results of each ranking were compared to the set of known aliases. Figure 3 shows the percentage of email address corresponding to the same entity as the source email address found in the top-k results for each alias. The smallest ranking included the top 10 likely alias candidates, whereas the largest ranking included 100 alias candidates. All rankings are out of a total of more than 14,000 email addresses.

It is possible for several email addresses to be ranked identically. Consider, the ranking based on geodesic distance assigns the same rank to all email addresses within the same distance from the source. If several email addresses are tied at the same rank in the results, the median position with the rank is used. For example, if 9 email addresses are tied at rank 1, an alias within these 9 emails would be reported as rank 5.

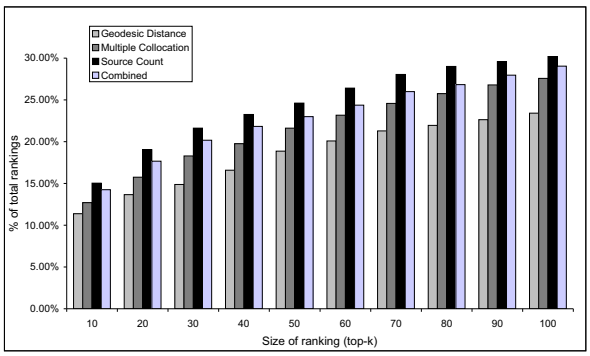


Figure 3: Percent of top-k rankings with at least one email address corresponding to the same entity as the source email address.

Figure 4 shows a comparison of the average precision-recall (PR) curves [2]. The ranking returned by a ranking method can be viewed as the result to a query, where the query is an email address and the result is a ranked list of possible alias candidates. Precision measures the fraction of results in the ranking which are relevant. Recall is the fraction of relevant items in the ranking which have been retrieved.

The PR curves were constructed using the rankings for all emails in the data set that have 6 aliases. The results for all cases have been averaged to produce an average precision and recall curve. Each level of recall represents one of the five aliases for each of these email addresses and therefore measures recall levels from 20% to 100%. Note, the precision for the combined ranking method, which incorporate both source size and multiple collocation heuristics, lies above the baseline (i.e. single heuristic and raw geodesic ranking) approaches for all levels of recall.

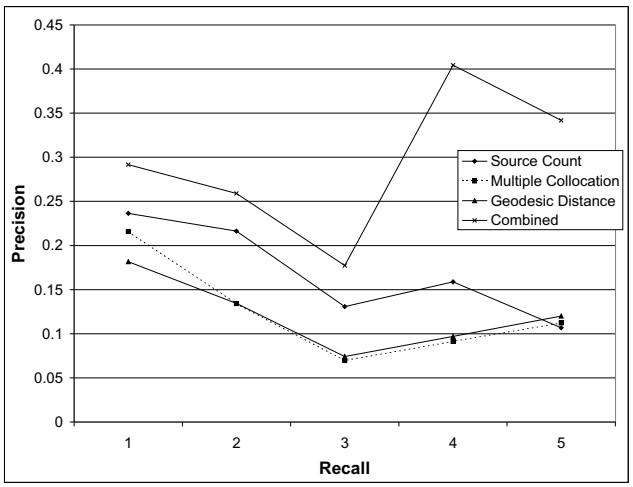


Figure 4: Average precision-recall curves for entities with 6 email addresses.

Figure 5 shows the number of predictions that were correctly identified at rank 1. From this figure it can be seen that all three heuristic methods described above perform better than picking one element from the email addresses at a geodesic distance of one. Specifically, the combined method almost doubles the probability of finding an alias at the first position in the ranking.

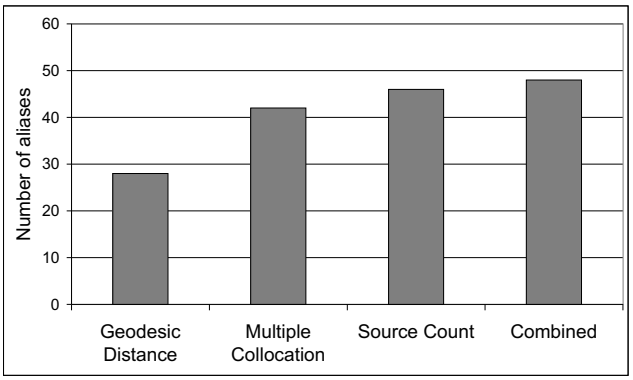


Figure 5: Number of aliases at rank 1.

5. DISCUSSION

This section discusses the results of the experimental evaluation above and addresses certain limitations of the applied methods.

5.1 Findings

The above analyses of email addresses in a social network setting demonstrates that most aliases tend to occur in close proximity of each other. Specifically, more than half of the email aliases are located within a geodesic distance of two from each other and about 20% of aliases are directly connected to each other. This confirms the hypothesis earlier regarding proximity and provides a basis for additional analysis.

The geodesic distance provides a useful method of locating a candidate list of aliases which correspond to the same entity. In more than 10% of the cases, an email alias can be found in the 10 closest addresses in the relational network. The methods developed for determining relationship strength are improvements upon geodesic-based rankings. First, Figure 3 shows that the multiple collocation heuristic is an effective method to increase the probability of finding an additional email aliases corresponding to the same entity. Second, making email address relationships inversely proportional to source size proves even more successful for increasing the probability of finding aliases in the results. Our results demonstrate that taking the number of email addresses on the page into account can increase the probability of finding an alias to more than 15% within the top 10 results, or a 0.0007 fraction of the total email addresses. It is interesting to note that the combined heuristic approach is not as effective as source size only, but does yield results better than geodesic distance.

Figure 5 demonstrates that each method improved the probability of finding an alias. In about 1.1% of the cases, an alias was found at rank 1. The use of the combined approach almost doubled the number of aliases to 2%. Even though the total number of aliases at rank 1 is small, all three methods using a heuristic measure for relationship strength significantly increased this number.

Figure 4 shows an average precision and recall curve for a small subset of aliases. This subset consisted of individuals with six email aliases. For each alias, the precision and recall curve was generated, by determining the rank at which each of the five remaining aliases where found in the total ranking of 14000 email addresses. The combined approach maintains a high level of precision over all levels of recall. The source size method also outperforms the simple geodesic method. Taking multiple collocation into account only showed minimal improvement. These results are mostly consistent with the other results.

5.2 Limitations and Improvements

Empirical results above demonstrate they are feasible in a controlled environment, such as a university, but a more thorough analysis is required. It is unclear how these algorithms will perform in a more general setting, such as the open Internet. One fundamental concern is that it is difficult to obtain a gold standard dataset. Thus unsupervised methods for evaluation must be designed.

Furthermore, there are many extensions to our detection methods which may increase success rate. Here, we briefly address several. First, a portion of the 14,000 email addresses studied correspond to non-human entities. One possible approach to correct this problem is to use a rule-based filter. Simple filter rules for common non-human users, such as “subscribe” or “feedback” may be simple and effective.

latanya.sweeney@cmu.edu
latanya@andrew.cmu.edu
latanya@cs.cmu.edu
latanya@lab.privacy.cs.cmu.edu
latanya@privacy.cs.cmu.edu

Table 4: Examples of email aliases with common id strings.

Second, usernames studied in the dataset are shared across the different domains, making it possible to determine each alias. Many individuals have multiple email addresses that share a common user id part. Table 5.2 depicts various email addresses for Latanya Sweeney in the Carnegie Mellon dataset. Note, though the subdomain changes, the string “latanya” is common to all email addresses. We do not expect that full names will remain constant across email addresses for the same entity, but we do expect there to be logical similarities. Along these lines, Bhattacharya and Getoor [4, 3] demonstrated that string comparator metrics [6], derived from the record linkage community, are feasible for relating name variants in social networks. As a result, we suspect that a comparison of the user id part of the email addresses in the ranked results would make it possible to determine a larger number of correct aliases.

6. CONCLUSION

This research demonstrated that email aliases corresponding to the same entity occur in close geodesic proximity within social networks inferred from online sources. While Geodesic distance provides a large candidate set of email addresses for a source email address, we show ranking methods can discover more precise sets by accounting for source size. Our results suggest that small numbers of email addresses collocated on the same web page are the most likely to have the strongest relationships. The alias detection methods correctly detect a significant number (i.e. better than random) of email addresses using only social relations. Though our methods are limited in precision at best rank predictions, we believe that improvements can be achieved through the incorporation of string comparator similarity metrics and rule-based filters.

7. ACKNOWLEDGEMENTS

The authors wish to thank Benoit Morel, as well as the members of the Data Privacy Laboratory for useful discussions and comments on this research.

8. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] R.A. Baeza-Yates and B.A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] I. Bhattacharya and L. Getoor. Deduplication and group detection using links. In *Proceedings of the ACM Workshop on Link Analysis and Group Detection (LinkKDD-2004)*, 2004.

[4] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2004.

[5] F. Ciravegna, A. Dingli, D. Guthrie, and Y. Wilks. Integrating information to bootstrap information extraction from web sites. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, Acapulco, Mexico, 2003.

[6] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[7] L. Cranor and B.A. Lamacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[8] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. *Proc. ACM/IEEE Joint Conf on Digital Libraries*, 2004.

[9] S. Hill. Social network relational vectors for anonymous identity matching. In *Proceedings of the IJCAI Workshop on Learning Statistical Models from Relational Data*, Acapulco, Mexico, 2003.

[10] P. Hsiung, A. Moore, D. Neill, and J. Schneider. Alias detection in link data sets. In *Proceedings of the International Conference on Intelligence Analysis*, McLean, VA, 2005.

[11] D. Kalashnikov, S. Mehotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 262–273, Newport Beach, CA, 2005.

[12] N. Kushmerick, E. Johnston, and S. McGuinness. Information extraction by text classification. In *Proceedings of the IJCAI Workshop on Adaptive Text Extraction and Mining*, Seattle,WA, 2001.

[13] B. Malin. Unsupervised name disambiguation via social network similarity. In *Proc. SIAM Wksp on Link Analysis, Counterterrorism, and Security*, pages 93–102, Newport Beach, CA, 2005.

[14] F. Mosteller and D.L. Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Reading, MA, 1964.

[15] J. Novak, P. Raghavan, and A. Tomkins. Anti-aliasing on the web. *Proceedings of the ACM World Wide Web Conference*, 2004.

[16] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. *Proceedings of Neural Information Processing Systems*, 2002.

[17] J.R. Rao and P. Rohatgi. Can pseudonymity really guarantee privacy? *Proceedings of the USENIX Security Symposium*, pages 85–96, 2000.

[18] L. Sweeney. Finding lists of people on the web. *ACM Computers and Society*, 34(1), 2004.

[19] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, New York, NY, 1994.

[20] S. White and P. Smyth. Algorithms for estimating relative importance in networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[21] W.E. Winkler. Matching and record linkage. In B.G. Cox, editor, *Business Survey Methods*. Wiley, New York, NY, 1995.

Capital and Benefit in Social Networks

Louis Licamele, Mustafa Bilgic, Lise Getoor, and Nick Roussopoulos
Computer Science Dept.
University of Maryland
College Park, MD
{licamele,mbilgic,getoor,nick}@cs.umd.edu

ABSTRACT

Recently there has been a surge of interest in social networks. Email traffic, disease transmission, and criminal activity can all be modeled as social networks. In this paper, we introduce a particular form of social network which we call a *friendship-event network*. A friendship-event network describes two inter-related networks. One is a friendship network among a set of actors. The other is an event network that describes events, event organizers and event participants. Within these types of networks, we formulate the notion of *capital* based on the actor-organizer friendship relationship and the notion of benefit, based on event participation. We ground these definitions in a real-world example of academic collaboration networks, where the actors are researchers, the friendships are collaborations, the events are conferences, the organizers are program committee members and the participants are conference authors. We incorporate a temporal component by considering the notion of an event series. We explore the use of these measures on a data set describing three computer science conferences over the past ten years.

1. INTRODUCTION

Recently there has been a great deal of interest in research involving social networks, including both modeling and analyzing the networks. A social network describes actors and their relationships and in some cases, events and actors' participation. A social network can be characterized by its relational structure; the underlying graph structure of the network dictates the structural properties. These include everything from the density of the graph and average degree of the nodes to the measure of centrality and information flow. Most of the research in social networks has focus on structural aspects of the networks.

In this paper we will look at networks that are a bit more complex than the classic 'who-knows-who' or friend-of-a-friend (FOAF) networks. In addition to friendship networks, we are also interested in event networks. Event networks include information about the organizers of an event and the participants in an event (these may be overlapping). We present a general formulation of these *friendship-event networks* (FEN).

To measure interesting structural properties of these networks, we define the notions of *capital* and *benefit*. Capital is a measure of an actor's social capital. It is defined in terms of the number of event organizers with whom an actor is friends. Benefit is defined from the perspective of an event organizer, in terms of how much benefit they give their friends and from the perspective of an event participant in terms of their participation in events. Depending on context, benefit may be perceived positively (as in the more benefit the greater the overall social capital of the network) or negatively (in terms of bias). Here we view them simply as descriptive properties useful for understanding the data.

Events naturally have a time associated with them and it is possible for relationships, positions and roles to change over time. These changes will in turn affect the social capital of an individual as well as benefit received and benefit given. To be more specific, events can occur at different times, the organizers of events change over time, and a different set of actors might participate in each event. In order to analyze temporal trends in capital and benefit properly, we must model these temporal aspects in our FEN.

To demonstrate the usefulness of the measures that we have developed, we apply them to academic collaboration networks. These networks describe researchers and their collaborations. We also have conference events along with their organizers (program committee (PC) members) and participants (authors) together which we will refer to as academic collaboration FEN. In this example dataset, a friend is defined as the people an author shares a co-authorship relation with, and social capital is the number of these friends who serve on the program committee for the conference in which the author publishes. Benefit given is expressed as the number of papers that the friends of a PC member publish in the conference, and benefit received is the number of publications that an author publishes in a conference.

We begin by describing some of the related work in Section 2. In Section 3, we give a general definition for the family of friendship and event networks that we study, and show the mapping to the academic collaboration networks. In Section 4, we define capital and benefit and in Section 5, we

LinkKDD'05 August 21, 2005 Chicago, IL, USA.

further extend our definitions with the important element of time. Finally, in Section 6 we describe some preliminary results applying these measures to three different computer science conferences over a 10 year time period.

2. RELATED WORK

A large portion of the work in mining social networks has focused on analyzing structural properties of the networks. For recent surveys, see Newman [11] and Jensen [6]. Much of the work has been descriptive in nature, but recently there has been more work which uses structural properties for prediction. Within this category, a number of papers focus on the spread of influence through the network (e.g., [4, 7]). These papers attempt to identify the most influential nodes in the network. Domingos and Richardson [4] use a global, probabilistic model that employs the joint distribution of the behavior over all the nodes. Kempe et al. [7] use a diffusion process that begins with an initial set of active nodes and uses different weighting schemes to determine whether or not a neighbor should be activated. McCallum et al. have proposed role discovery in social networks by looking at messages sent and received between entities [10]. Liben-Nowell and Kleinberg [8] attempt to predict future interactions between actors using the network topology. In addition, Palmer et al. [12] propose an efficient method for approximating the connectivity properties of a graph.

Even though social capital is defined slightly differently in different contexts such as sociology and economics, most definitions agree that social capital is a function of ties between actors in a social network whereas human capital refers to properties of individual actors. Degenne and Forse [3] trace the idea back to Hobbes who said "to have friends is power" [5]. However, the term itself and its systematic studies are relatively recent [1, 9, 2]. Portes argues that a systematic treatment of social capital must distinguish between the "possessor of the capital" (actors who receive benefits), "sources of the capital" (actors who give benefits), and the resources that have been received or given [13]. In our analysis, the "sources of the capital" are the organizers of the events. Two related notions in social network analysis are position and role; position refers to subsets of actors who have similar ties to other actors, and role refers to patterns of relationships between these actors or subsets [14].

3. THE FRIENDSHIP-EVENT NETWORK

We begin with a generic description of a family of social networks which we refer to as FEN. A FEN has the following sets of entities:

- **actors:** a set of actors $A = \{A_1, \ldots, A_n\}$
- **events:** a set of events $E = \{E_1, \ldots, E_m\}$

and the following sets of relationships:

- **friends:**

$$F(A_i, A_j) = A_i \text{ is friends with } A_j$$

- **organizers:**

$$O(E_k, A_i) = A_i \text{ is an organizer of event } E_k$$

- **participants:**

$$P(E_k, A_i) = A_i \text{ is a participant in event } E_k$$

We use $f(A_i)$ to denote the friends of actor A_i, i.e.,

$$f(A_i) = \{A_j \mid F(A_i, A_j)\},$$

and $o(E_k)$ to denote the organizers of event E_k, i.e.,

$$o(E_k) = \{A_i \mid O(E_k, A_i)\},$$

and $p(E_k)$ to denote the participants in event E_k, i.e.,

$$p(E_k) = \{A_i \mid P(E_k, A_i)\}.$$

In some cases, it makes sense to allow an actor to participate in an event more than once. In these cases, for each E_k, we define an associated set of subevents,

$$se(E_k) = \{e_{k1}, \ldots e_{kp}\},$$

and define a participant subevent relation:

$$S(E_k, A_i, e_{kj}) = A_i \text{ is a participant in subevent } e_{kj} \text{ of } E_k$$

Then the participants can be defined in terms of the subevent relation:

$$P(E_k, A_i) = \exists\ e_{kj} \in se(E_k) \text{ s.t. } S(E_k, A_i, e_{kj})$$

In terms of the academic collaboration example, the actors are the researchers (both authors and PC members) and the events are the conferences. The friendship relation is defined based on whether two researchers have co-authored a paper together. In this case the friendship relationship is symmetric, but this may not be true in other domains. The organizers of an event are the PC members and the participants in the event are the set of authors that have papers published in the conference. Since authors may have more than one publication in a conference, the subevent relationship is authorship of a paper (the subevent) in a conference. An illustration of the academic collaboration FEN is given in Figure 1.

4. EVENT-SPECIFIC CAPITAL AND BENEFIT

Next we introduce the notions of capital and benefit. *Personal social capital* is a measurement of the amount of "goodwill" available to an actor based on the actor's friendship relationships. We begin by defining social capital in the context of a single event E_k.

Definition 1. Social Capital: The personal social capital of an actor A_i in an event E_k is the number of organizers with whom the actor is friends:

$$SC(A_i, E_k) = \sum_{A_j \in o(E_k)} I(F(A_i, A_j))$$

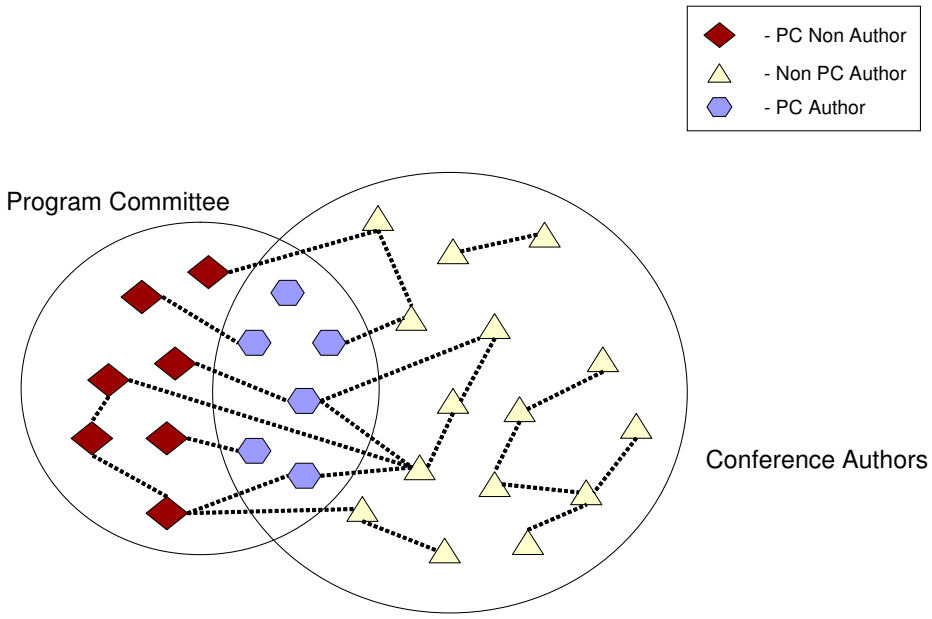


Figure 1: An event in the friendship-event network for academic collaboration. The actors in the network are PC members and authors. The edges in the network indicate co-authorship links (friendship). The organizers are the PC members (the set on the left), and the participants are the authors (the set on the right). Note that these sets need not be disjoint; i.e. a PC member can be an author as well. The three categories of actors are: PC-Non-Authors, PC-Authors, and Non-PC-Authors. If we name the sets as PC and CA from left to right, these categories refer to the sets $PC \setminus CA$, $PC \cup CA$, and $CA \setminus PC$ respectively.

where I is an indicator function which is 1 when the relation holds. [1]

The definition is based on Hobbes's idea that it is more important to have powerful friends than to have numerous powerless friends [5]. Therefore, we define an actor's capital in terms of organizer friends rather than simply friends. We also define the notion of the *Social Capital Ratio* which is the proportion of the organizing committee with whom an actor is friends:

Definition 2. Social Capital Ratio: The personal social capital ratio of an actor A_i in an event E_k is the proportion of organizers with whom A_i is friends:

$$SCR(A_i, E_k) = \frac{\sum_{A_j \in o(E_k)} F(A_i, A_j)}{|o(E_k)|}$$

Next we turn to a definition of *Benefit*. We can look at benefit from both the perspective of an event participant and an event organizer. In our model, participation in an event is considered beneficial. As mentioned earlier, we may consider participation to be a binary yes/no relationship, or, alternatively, actors may participate in an event more than once, and the more an actor participates, the more benefit they receive. Given our motivating example, the latter definition is more appropriate, so we use it in our definition of benefit below.

Definition 3. Benefit Received: Actors receive benefit when they participate in events. The benefit received by an actor A_i in event E_k is:

$$BR(A_i, E_k) = \sum_{e_{kj} \in se(E_k)} S(E_k, A_i, e_{kj})$$

In the context of the academic collaboration FEN the benefit an author receives for a given conference is the number of publications the author has in the conference. We also define the benefit received ratio as the proportion of conference paper authorships (where a paper with 3 authors counts as 3 paper authorships):

Definition 4. Benefit Received Ratio: The benefit received ratio for an actor A_i in event E_k is:

$$BRR(A_i, E_k) = \frac{BR(A_i, E_k)}{\sum_{A_j \in A} BR(A_j, E_k)}$$

From the perspective of an event organizer, we measure the benefit given. Benefit given is the benefit that an event organizer's friends receive.

[1] To improve readability, we will drop the I in the definitions that follow, but throughout the intended interpretation is that we are counting the number of times a relation or expression holds.

Definition 5. Benefit Given: The benefit given by an organizer A_o of an event E_k is:

$$BG(A_o, E_k) = \sum_{A_i \in f(A_o)} BR(A_i, E_k)$$

and the benefit given ratio is the percentage of all conference benefit that an organizer is responsible for:

Definition 6. Benefit Given Ratio: The ratio of benefit given by an organizer A_o of an event E_k is:

$$BGR(A_o, E_k) = \frac{BG(A_o, E_k)}{\sum_{A_i \in o(E_k)} BG(A_i, E_k)}$$

We can also look at benefit from the event perspective by aggregating these measure over events:

Definition 7. Average Benefit Received Ratio and Average Benefit Given Ratio: The average benefit received ratio for an event E_k is:

$$ABRR(E_k) = \frac{\sum_{A_i \in p(E_k)} BRR(A_i, E_k)}{|p(E_k)|}$$

And the average benefit given ratio for an event E_k is:

$$ABGR(E_k) = \frac{\sum_{A_o \in o(E_k)} BRG(A_o, E_k)}{|o(E_k)|}$$

5. TEMPORAL ASPECTS

Social networks are dynamic so time obviously plays an important role. We look at two temporal components to our FEN.

5.1 Event Series

It is often the case that there is not just a single event, but that multiple events form an event series. The conferences in our academic collaboration FEN are one example, but others include regularly scheduled meetings, a book or movie series or a series of sporting events.

We introduce the notion of an event series by adding a time index to our events:

- **event series:** an event series $E_k(T)$ is composed of a set of events $E_k(t_1), \ldots, E_k(t_q)$

The notions of benefit received and benefit given defined above can easily be extended to event series. For example, the overall average benefit received ratio for a conference series $E_k(T)$ is:

$$OABRR(E_k(T)) = \frac{\sum_{t=t_1}^{t_q} ABRR(E_k(t))}{q}$$

and similarly we can define $OABGR(E_k(T))$, the overall average benefit given ratio for a conference series $E_k(T)$.

Table 1: For each conference series, the average number of papers, average number of authors and average PC size for the past 10 years.

Conf.	Papers		Authors		PC	
	μ	σ	μ	σ	μ	σ
C1	78.90	9.45	223.20	25.24	32.60	5.87
C2	87.00	23.75	237.70	85.89	69.62	23.30
C3	29.20	2.94	66.30	9.87	9.30	2.87

5.2 A Temporal Definition of Friendship

Now that we have a notion of time associated with events, clearly we must update our definition of friendship so that we only consider current friends and not future friends in our calculations. We modify the definition of friendship to include a temporal argument: $f(A_i, A_j, t)$ means that A_i and A_j are friends at time t. In the case of our academic collaboration FEN, we say that A_i and A_j are friends at time t if they co-authored a paper which was published at or before time t.

Sadly enough, friendships may fade over time. In addition to the above definition which defines friendship at a particular time, we also introduce a time window, which allows us to consider only friendships within a certain recency window. For the academic collaboration FEN, we say that A_i and A_j are friends at time t if they co-authored a paper which was published within a time window of size n before time t.

Definition 8. Friendship: Two authors are considered friends at time t if they have co-authored a paper within last n years.

$$F(A_i, A_j, t, n) \Leftarrow \exists t' CoAuthor(A_i, A_j, t') \wedge 0 \leq t - t' \leq n$$

6. EXPERIMENTAL RESULTS

We explored how these descriptive statistics apply to a real academic friendship-event network. We measured friendship, capital and benefit on a dataset describing publication information and program committee members for five major conferences of a subfield of computer science. There are 11,644 unique papers from 1959 till 2004, and these papers contain 11,554 unique authors. There are 1,821 distinct program committee members. Because two of the conferences have missing data for PC members, we leave them out for the capital and benefit analysis, but use their publications for defining friendships.

The summary statistics for the data are given in Table 1. The μ and σ are computed for the last 10 years of the data, i.e. from 1994 to 2003. As we can see, C1 and C2 can be considered similar in terms of having a relatively large number of papers, a large number of authors and relatively large PC. C3 on the other hand, is significantly smaller. It turns out that C1 and C2 are two flagship conferences for the area, and are more applied, while C3 is has a more theoretical bent.

Our measures are *not* calculating bias in paper acceptance. There are many reasonable explanations for why there should be correlations in the measures we have defined, for example in certain communities PC members may be more likely

Table 2: Overall aggregate statics for friendship, capital, benefit given (BG) and benefit received (BR).

Conf.	Friendship		Capital		BR		BG	
	μ	σ	μ	σ	μ	σ	μ	σ
C1	8.29	2.50	0.55	0.20	1.16	0.46	4.64	10.28
C2	7.45	1.20	0.71	0.42	1.09	0.34	3.13	8.41
C3	8.37	2.31	0.57	0.29	1.10	0.33	3.15	4.64

to be younger, tenure-track academics under greater pressure to publish, while in other communities PC members may be more senior, with larger and more productive research groups. Further, because we do not have complete data, our results are at best approximations to the measures we have defined. In particular, we do not have information about rejected papers. Additionally, we do not know the reviewer assignments, so when an author submits a paper to a conference, we do not know who reviews her paper. Specifically, we do not know if an author has been assigned a reviewer whom is also a friend. We also do not have access to the reviews, so we do not have a measure of quality assigned to the paper. Even if the author's friends review the paper, we do not know if the paper was accepted because the paper was of good quality or as a result of a favor. Our notion of benefit therefore is *not* capturing unfairness in the reviewing process. Nonetheless, we believe that the notions that we have introduced are useful descriptive measures for friendship-event networks. And as far as we are aware, their quantitative definitions are novel.

Overall aggregate statistics for the conferences are shown in Table 2. Here we are using a friendship window size of 5 years (i.e. $n = 5$). Interestingly, despite the difference in the sizes of the friendship-event networks for the three conferences, the aggregate structural statistics are surprisingly similar. The statistics are not significantly different for all three conferences; the means are all less than one standard deviation away from one another. The only significant difference is in the standard deviations in benefit given (BG) for conference C1 and C2 as compared to conference C3. These statistics imply that the same phenomenon is present in all three conferences.

6.1 Role-based Comparison

In order to analyze the data in further detail, we broke the actors into three groups according to their roles in the network. For a particular conference and year, we have the following sets of actors:

- **PC-Authors:** Program committee members who have also published in that conference in that year.
- **PC-Non-Authors:** Program committee members who have not published in that conference in that year.
- **Non-PC-Authors:** Authors who are not in the program committee but published a paper in that conference in that year.

We analyzed the data to see if there are any apparent differences between these groups in terms of either capital or benefit. We began by looking at the different classes of authors, PC-Authors and Non-PC-Authors. Table 3 show the average benefit received for each of these groups. The average benefit received is not significantly different between these two groups. This reflects the fact that most people only have one paper in the conference. It could be interesting to further study how the capital of these two groups effects the number of publications of each author.

Table 3: Average Benefit Received for PC-Authors versus Non-PC-Authors.

Conf.	PC-Author		Non-PC-Author	
	μ	σ	μ	σ
C1	1.38	0.64	1.15	0.45
C2	1.26	0.58	1.08	0.31
C3	1.21	0.45	1.10	0.33

Table 4: Capital for PC-Authors and PC-Non-Authors

Conf.	PC-Author		PC-Non-Author	
	μ	σ	μ	σ
C1	1.29	0.09	0.20	0.05
C2	1.76	0.24	0.27	0.01
C3	0.72	0.10	0.11	0.05

We next shift our focus to determine what makes PC-Authors and PC-Non-Authors different. We start by examining how social capital differs between these groups. The average capital for each group is reported in Table 4. We see that the average capital is significantly higher for PC-Authors compared to Non-PC-Authors. So PC members are more likely to publish in their own conference if they have a lot of friends on the program committee with them. There are several potential explanations for this difference. Perhaps these PC members are benefiting from having many friends on the program committee. On the other hand, it may be that having a lot of fellow PC friends is an indication of how well-suited the conference is to the PC-Author's research. Alternatively, it may be that PC members with a lot of fellow PC friends, are more inclined to submit because they want to attend the conference with their friends. Of course we cannot draw concrete conclusions from this one insight, but it can help to further guide our understanding of this social network.

6.2 Event Series Analysis

As has been shown, the notion of friendship and capital can allow for insights to be made when comparing different conferences. We have been aggregating the values over a

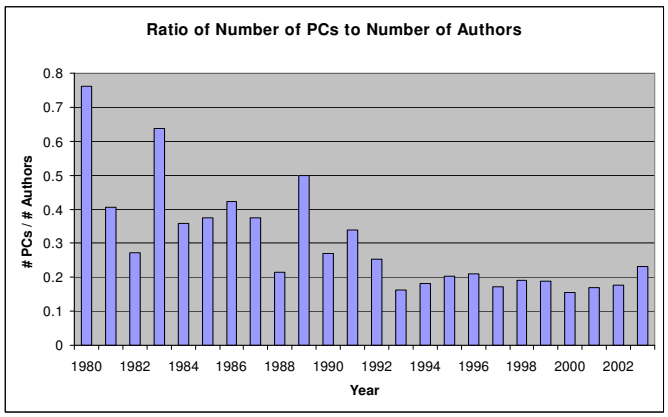


Figure 3: PC Author Ratio

ten year period to allow for a comparison of conferences as a whole, but we can also look at trends in a specific conference. The patterns of a conference over time can be shown by inspecting how friendship and capital change throughout the years. We present a more detailed inspection of conference C1 in order to demonstrate how these new notions can help for this exploratory data mining process.

One of the first things that someone might be interested in finding out is how the levels of friendship compare between the different categories of actors. This information was calculated for the last 23 years of conference C1 and is shown in Figure 2(a). The temporal trends of the levels of friendship among all of the categories is shown in this figure. The friendship levels increase over time. These values are an average over all the individuals involved, so it is not skewed by the increase in the number of authors or the size of the program committee over time.

The amount of friendship for the PC-Authors is one of the first things that stands out in this graph. It appears that the PC-Authors have more than double the number of friends than both the Non-PC-Authors and the PC-Non-Authors. One explanation for the difference in friendship between PC-Authors versus Non-PC-Authors is that we might assume that PC members have more friends and that is why they are chosen to be on the program committee. In that case, we would expect for the friendship values of all PC members, not just the ones who are authors but also the PC-Non-Authors, to be higher than the friendship of the Non-PC-Authors. The PC-Non-Authors have a slightly higher friendship value than the Non-PC-Authors but it is still a much smaller friendship value than what the PC-Authors have. So we can see that, on average, PC members have more friends than Non-PC members.

To better understand these differences, we examined the amount of capital of each group. Given the number of friends that a person has, and assuming that each friend had an equal chance of being on the program committee, we would expect to find similar patterns in the capital values between the groups as was shown in Figure 2(a). The capital values are shown in Figure 2(b). The same overall upward trend that was seen for the friendship values is present. The PC-Authors' social capital is still more than double the values of the other two groups. Of course, in many ways this is not surprising because they had the most number of friends. The interesting results in this graph are those of the PC-Non-Authors. Though it was shown that they have more friends than Non-PC-Authors, it appears that they have less friends on the program committee.

Another way to look at the difference in trends between the friendship and capital values is to examine the ratio of capital over friendship. This ratio is shown in Figure 2(c) for all actor groups. Overall, the PC-Authors have the highest percentage of friends that are on the program committee with them. The Non-PC-Authors have the next highest percentage of friends on the program committee. The PC-Non-Authors have a much lower percent of their friends that are on the program committee. Maybe this is why they are much less likely to publish in the conference that they are the PC on.

It is hard to draw conclusions from just the changes in friendship and capital alone. It could be possible for outside variables to affect these two values. One possible scenario that would lead to an increase capital over time would be if the size of the program committee increased each year, which in many cases it does. To check if this trend exists in this data, we calculate the ratio of total number of program committee members to the total number of authors per year. These results are presented in Figure 3. As it turns out, the size of the program committee grows at a slower rate than the total number of authors overall and over the last ten years this ratio has stayed somewhat static.

7. FUTURE WORK

We would like to examine richer notions of friendship. The friendship relationship is currently a boolean feature. If two actors are related, or in the scenario presented here if they have co-authored a paper together, within the last n years, we consider them friends. Alternatively, we can formulate friendship as a function that maps to a real number, which monotonically decreases as the relationship ages without reinforcement. That is, we can consider two authors' friendship to be stronger than another pair's friendship if the former pair had published a paper together more recently. Moreover, we can also take the number of times two authors published together as an indicator of the strength of the friendship as well; the more they publish together, the stronger the relationship they have. It might be interesting to explore the effects of these new formulations of friendship on the statistics.

8. CONCLUSION

We have formulated a general family of friendship-event networks. We have given quantitative definition for social capital, benefit recieved, and benefit given. At this point, our analysis is purely descriptive; we are interested in measures that help us understand friendship-event networks and that allow us to compare different event series. Ideally, these definitions could be used as part of a design process, that would, depending on the context allow us to construct friendship-event networks that would either maximize or minimize benefit. This could be of use for a variety of tasks such as

constructing program committees, assigning reviewers and author networking.

9. REFERENCES

[1] P. Bourdieu. *Handbook of Theory and Research for the Sociology Education*, chapter The forms of capital, pages 241–258. Greenwood, New York, 1985.

[2] J. S. Coleman. Social capital in the creation of human capital. *The American Journal of Sociology*, 94:S95–S120, 1988.

[3] A. Degenne and M. Forse. *Introducing Social Networks*. SAGE Publications, London, 1999.

[4] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD*, pages 57–66, New York, NY, USA, 2001. ACM Press.

[5] T. Hobbes. *Leviathan*. Collier, New York, 1962.

[6] D. Jensen and J. Neville. Data mining in social networks. In *National Academy of Sciences Symposium on Dynamic Social Network Analysis*, 2002.

[7] D. Kempe, J. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD*, pages 137–146, New York, NY, USA, 2003. ACM Press.

[8] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Intl. Conf. on Information and Knowledge Management*, 2003.

[9] G. C. Loury. Intergenerational transfers and the distribution of earnings. *Econometrica*, 49(4):843–867, July 1981.

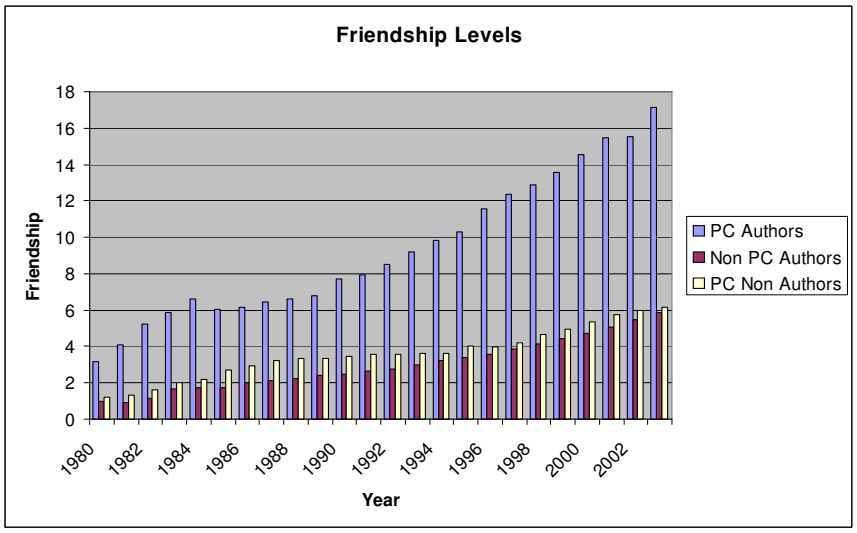
[10] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI '05: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 2005.

[11] M. Newman. The structure and function of complex networks. *IAM Review*, 45(2):167–256, 2003.

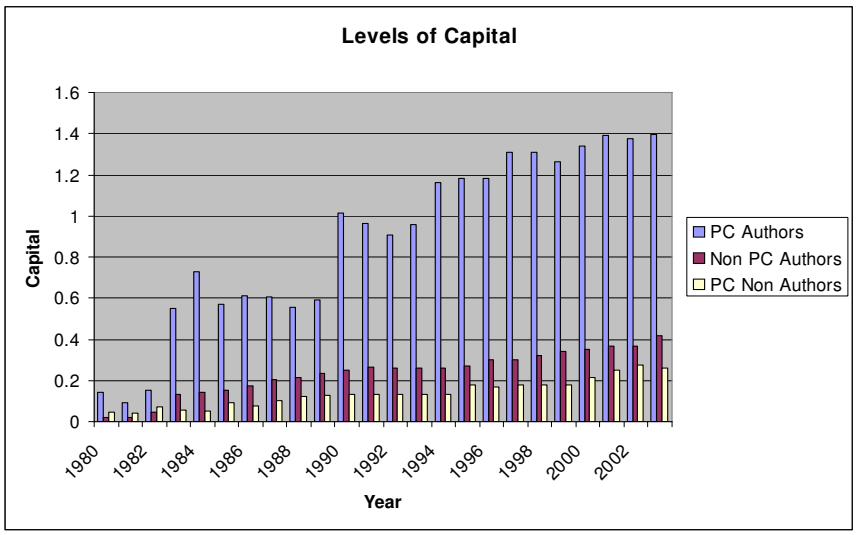
[12] C. Palmer, P. Gibbons, and C. Faloutsos. ANF: A fast and scalable tool for data mining in massive graphs. In *ACM Intl. Conf. on Knowlege Discovery and Data Mining*, 2002.

[13] A. Portes. Social capital: Its origins and applications in modern sociology. *Annual Review of Sociology*, 24:1–24, 1998.

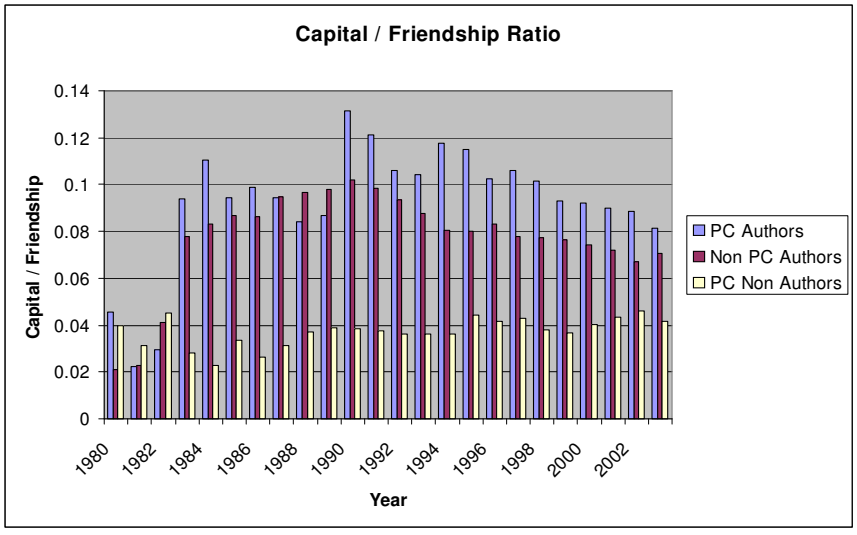
[14] H. C. White, S. A. Boorman, and R. L. Breiger. Social structure from multiple networks: I. blockmodels of roles and positions. *American Journal of Sociology*, 81:730–779, 1976.



(a)



(b)



(c)

Figure 2: Detailed analysis of C1 over 10 years (1994-2003) for (a) friendship (b) capital and (c) captial-friendship ratio

EventRank: A Framework for Ranking Time-Varying Networks

Joshua O'Madadhain (jmadden@ics.uci.edu)
Department of Computer Science
University of California, Irvine

Padhraic Smyth (smyth@ics.uci.edu)
Department of Computer Science
University of California, Irvine

ABSTRACT

Node-ranking algorithms for (social) networks do not respect the sequence of events from which the network is constructed, but rather measure rank on the aggregation of all data. For data sets that relate to the flow of information (e.g., email), this loss of information can obscure the true relative importances of individuals in the network. We present EventRank, a framework for ranking algorithms that respect event sequences and provide a natural way of tracking changes in ranking over time. We compare the performance of a number of ranking algorithms using a large organizational data set consisting of approximately 1 million emails involving over 600 users, including an evaluation of how the email-based ranking correlates with known organizational hierarchy.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database applications—*data mining*; G.2.2 [**Discrete Mathematics**]: Graph theory—*network problems*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms

Algorithms, Experimentation, Measurement, Verification

Keywords

Network ranking algorithms, network temporal evolution, social network analysis

1. INTRODUCTION AND MOTIVATION

There exist a variety of algorithms that rank entities in a social network according to criteria that reflect structural properties of the network. These criteria are generally intended to measure the "influence", "authority", "prestige", or "centrality" of the entities in the community represented by the network. Examples of such ranking algorithms include betweenness centrality [2], eigenvector centrality [9] (and related algorithms such as PageRank [3]), HITS [6], and voltage-based rankers [12].

LinkKDD'05 August 21, 2005, Chicago, IL, USA

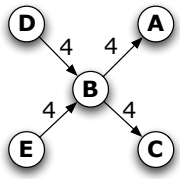


Figure 1: A network representing a sequence of messages

Each of these algorithms makes the implicit assumption that the network is a static object, i.e., that the entity and relationship sets, and the rank of each in the community, do not change over time. However, in some cases, this assumption is clearly false [4]. Examples include research citation networks (researchers may gain prestige over time if they publish papers that are cited by many people, and may lose it if they stop publishing), and email networks (correspondents' participation in email, and thus the extent to which they are "in the loop", may change on several time scales, depending on such factors as patterns of email access, vacations, and changes of status in an organization).

It is, of course, possible to repeatedly apply one of the above ranking algorithms to successive "snapshots" of the data (that is, subsets of the data restricted to a particular interval) to yield a sequence of rank values that vary over time [5]. However, in the context of data sets that relate to the flow of information (e.g., email), the sequence of events can be significant in determining the relative importances of individuals in the network, and this information is lost when events are aggregated into a single snapshot of the network. Thus, rank values on networks which represent an aggregation of data over time can be thought of as representing summary statistics (e.g., sums or means) of the ranks over time.

One can also, given a static picture of a network, assign weights to edges that reflect the amount of elapsed time since each associated event occurred [13]. However, while this may yield a better model of the edge weights at a given time than a simple summation of the number of events in which the individuals mutually participated, it still fails to capture the information represented by the sequence of messages.

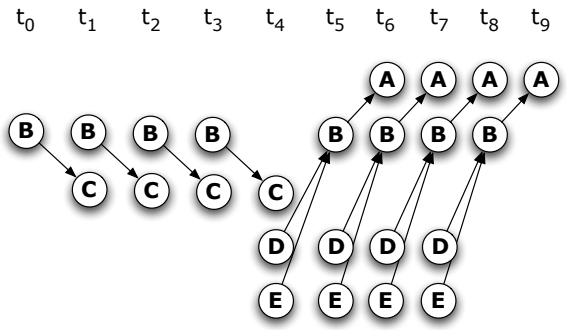


Figure 2: A message sequence that could have resulted in the network in Figure 1

For an illustration of this information loss, consider the example network shown in Figure 1, in which the directed edge $< X, Y >$ exists if X has emailed Y, and the value associated with $< X, Y >$ denotes the number of messages that X has sent to Y. Any of the ranking algorithms mentioned thus far would conclude that the ranks of A and C were the same; in the terminology of social network analysis, A and C are structurally equivalent.

However, the sequence in which email is sent carries information about the underlying communication of information; we can model correspondents as having a state, which changes in response to the receipt (and possibly the sending) of a message. Suppose that we have two emails e_{XY} from X to Y, and e_{YZ} from Y to Z; we denote the time of an email by $t(e_{UV})$. If $t(e_{XY}) < t(e_{YZ})$, then the content and timing of e_{YZ} may reflect the state change caused by e_{XY}. However, if $t(e_{XY}) > t(e_{YZ})$, then e_{YZ} cannot reflect any information contained in e_{XY}.

Figure 2 represents one possible sequence of messages that could have resulted in the collapsed network shown in Figure 1; there are, of course, many such sequences. This sequence suggests that, at time t_9, A should be considered to be more important than C, both because A has been a message participant more recently, and because the sequence suggests that A may be receiving information from D and E as well as from B. It is also important to point out that at time t_4 the opposite is true (that is, we would consider C to be more important than A).

We will discuss two different types of measures for temporal ranking: *transient*, which is a measure of the current rank at a particular time t, and *cumulative*, which is a measure of rank that encompasses the interval $[t_0, t]$.

In this paper, we will describe a framework for such measures. Our examples and model will focus specifically on email traffic data, but we believe that this framework may have wider application to ranking entities in data sets which consist of sequenced events that induce (or reflect) a network of relationships.

2. MODEL

We can model the functioning of algorithms such as PageRank or the voltage ranking algorithm as the flow of "potential" in a network from each entity to neighboring entities; possession and/or transfer of this potential is the basis for these measurements of rank. This potential flow can be modelled by repeated multiplication of the vector of original potential values (generally a uniform distribution) by a matrix M which represents the network.

We borrow this metaphor of potential flow to describe the functioning of the models in our framework: the potential values at time t_{i+1} may be calculated based on those at time t_i by multiplying the potential vector by a matrix M_i which represents the effect of the message sent at time t_i. Thus, transient rank may be defined as the amount of potential present at time t, whereas cumulative rank may be defined in terms of the mean potential value for the interval $[0, t]$.

There are two key distinctions between existing models and those arising from the framework that we propose. First, algorithms such as PageRank generate rankings that correspond to a stationary distribution of potential over entities, but by design, our models generate ranks that do not converge to a single value, because the matrices M_i, which represent messages with different senders and recipients, are not all identical. Second, algorithms such as PageRank use potential flow as a model of a random traversal of the network; by contrast, our models use potential flow to model exactly those transitions which correspond to the events for which we have evidence, in the sequence in which the events occurred: potential flows if and only if a message is sent.

The definition of rank in a social network is generally somewhat subjective; ranking algorithms generally do not have a "ground truth" to which their output can be compared to determine accuracy of the ranking model, although ranks for smaller social networks can be validated in part by comparing their results with those of surveys of entities in the network. As such, the validity of a ranking model is generally evaluated first in terms of its axiomatic properties. While not quite a complete axiomization, we nonetheless present here a list of desiderata that we believe should be satisfied by any model whose purpose is to calculate entity ranks based on email traffic.

Note that in the list below, a message *participant* is either a sender or a recipient of a specific message m; all other entities are *non-participants* of m.

1. Ranks should be comparable across time. (Ideally this would mean that at each step that ranks are automatically normalized, but at least they should be able to be normalized at any time.)
2. Receiving a message from an individual of rank r should lead to an increase in rank at least as large as that from receiving a message from an individual of rank $q < r$, all other things being equal.
3. Sending a message should not fail to have an effect because the sender has no "potential".
4. The ranks of the participants of m should not decrease in response to m. (There might be circumstances in which sending and receipt should have no effect on participants' ranks.)

5. The ranks of the non-participants of m should not increase in response to m. (Again, it may be permissible for non-participants' ranks to remain constant.)

6. Rank value evolution should be sensitive to message sequence.

These requirements might appear more stringent than those of other ranking algorithms, but this is essentially a reflection of the fact that we wish to model the effect of individual successive events (in this case, emails) on rank values, as opposed to modeling the effects of all such events in parallel.

The potential flow for a message in this framework will take the following general form: the non-participants send some of their potential to the sender; the sender retains some fraction of this potential (which causes the sender's potential to increase) and distributes the rest among the recipients (which causes each of the recipients' potential to increase). This scheme satisfies each of the requirements enumerated above: potential is conserved, which means that the transient ranks are automatically normalized (and thus comparable across time); the sender always has potential to send (unless there are no non-participants, in which case the message is effectively spam and should have no effect on anyone's rank); the participants gain potential (or at least lose none); and the non-participants lose potential. If we process messages in chronological order, then models in this framework will automatically satisfy all our stated requirements.

We denote the potential of correspondent $c \in C$ at time t_i by $R_i(c)$, which takes on values in the interval $(0, 1)$. $R_0(c) \equiv \frac{1}{|C|}$, and in general $R_i(c)$ is recursively defined as

$$c \in P_i \quad : \quad R_{i-1}(c) + \alpha_i \cdot \frac{\bar{R}_{i-1}(c)}{\sum_{d \in P_i} \bar{R}_{i-1}(d)} \tag{1}$$

$$c \notin P_i \quad : \quad R_{i-1}(c) \cdot \left(1 - \frac{\alpha_i}{T_{N_{i-1}}}\right) \tag{2}$$

where m_i is the message sent at time i, P_i is the set of participants of message m_i, α_i is the total amount of potential that the message m_i contributes to the participant set, $\bar{R}(d, t_i)$ is the additive inverse of d's potential, i.e., $1 - R_i(d)$, and $T_{N_{i-1}}$ denotes the total amount of potential held by the non-participants of m_i, that is, $\sum_{d \notin P_i} R_{i-1}(d)$.

The α_i values characterize the potential values' volatility–that is, larger values indicate that non-participants retain less of their potential–and are constrained as follows:

$$0 \leq \alpha_i \leq T_{N_{i-1}} \tag{3}$$

The definition of α_i may depend on a number of factors, such as the size of P_i relative to $|C|$, the elapsed time since the most recent message that the sender received from any of the recipients, the number of messages that the recipients have sent to the sender for which replies are pending, the elapsed time since the most recent message that the sender has sent, and so forth. Note that if $\alpha_i = 0$, then the potential values do not change at time step t_i, and if $\alpha_i = T_{N_{i-1}}$, then the potential values of the non-participants go to 0 (and further transfer of potential in response to subsequent messages will not occur as long as their participant sets are equal to P_i).

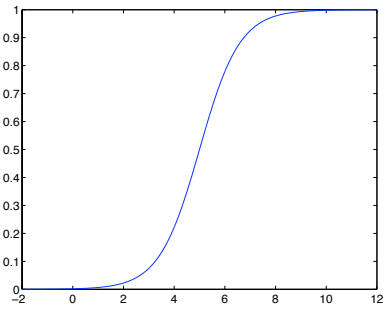


Figure 3: $g(\Delta t_s, 5)$

We observe that, in general, as P_i grows, the total amount of potential available ($T_{N_{i-1}}$) decreases; also, the changes in potential value to $d \in P_i$ decrease (because α_i is smaller, and because the number of correspondents among which α_i is divided is larger). In particular, if $P_i = C$, then m_i does not result in any transfer of potential. This ensures that messages with wide distribution ("spam") have little or no effect on potential.

The potential α_i is distributed among the elements of P_i in proportion to the additive inverse of their potential values: thus, the lower the potential of a participant, the more potential is assigned to it. (Senders and recipients, in this portion of the model, are treated equivalently; a more complex model might give senders or recipients more "credit" for their participation.)

In this paper, we explored two models for defining α_i. In the first ("baseline"), α_i was set to a constant fraction $f \in (0, 1)$ of $T_{N_{i-1}}$ for various values of f:

$$\alpha_i = f \cdot T_{N_{i-1}} \tag{4}$$

The second ("reply") model for α_i, elaborates the baseline model for α_i by discounting it according to functions of two factors: Δt_s (the elapsed time since the last message sent by the sender of m_i), and Δt_r (the elapsed time since the last message received by the sender from any recipient of m_i):

$$\alpha_i = f \cdot T_{N_{i-1}} \cdot g(\Delta t_s, G) \cdot h(\Delta t_r, H) \tag{5}$$

where g and h both take on values in the interval $(0, 1)$.

We define g as

$$g(\Delta t_s, G) = \frac{\tanh(\frac{10 \Delta t_s}{G \cdot \pi} - \pi) + 1}{2} \tag{6}$$

where G is a positive constant that specifies the amount of time required for a sender to "recharge" to the point that her next message will have half the maximum possible effect. Figure 3 shows that this functional form for g guarantees that its output increases with Δt_s, while being restricted to the range $(0, 1)$. This modification is motivated by our desire to prevent individuals who send messages much more frequently than the norm (specified by G) from dominating the rankings.

We define h as

$$h(\Delta t_r, H) = 2^{-\frac{\Delta t_r}{H}} \tag{7}$$

where H is a positive constant that specifies the "half-life" of a message (the amount of time required for the effect of a reply to drop to half the maximum); this boosts α_i for messages that are quick replies to other messages. (Since we do not have access to the email headers in the experimental data used below, we assume that a message sent from s to a set of recipients D is a "reply" if any $c \in C$ has sent a message to s since the last message that s sent to D.) This refinement to the model reflects observations that have been made to the effect that the speed of response to email can carry information [11].

It is possible to model G and H as functions of the characteristics of individual participants, or of pairs of individuals, rather than as global features; H, in particular, is likely to depend in practice on individuals' attitudes regarding email etiquette. However, we do not have enough information for the data set used in this paper to be able to model individual characteristics in this way.

We note that there are many other plausible approaches to defining a set of update equations and parameters and we do not claim that the specific methods proposed above are in any way unique or optimal. Later for example we will look at the sensitivity of the results to whether or not a reply component is included in the model, the setting of the parameter f, and so forth.

The time complexity of handling a single message is nominally $O(n)$; however, we can increase the efficiency by lazily updating the potential values of non-participants (that is, only updating the potential values of P_i at time t_i). We do this by storing (a) the sequence of α_i values, and (b) for each correspondent c, the index of the last message for which c was a participant. We then can apply all "skipped" α_i values at once when the next message for which c is a participant is processed (or after all messages have been processed, if there is no such message).

Based on this model, we define the following measures of cumulative rank for a correspondent c: sum of "outgoing" potential (that is, changes to c's potential caused by c sending a message), sum of "incoming" potential, and sum of transient ranks (that is, sum of potential values at each step); we refer to these hereafter as S_o, S_i, and S_r respectively. Note that S_o and S_i are analogous to the HITS "hub" and "authority" scores, respectively, or to outdegree and indegree.

3. EXPERIMENTS

Our experiments were performed on approximately 1 million emails spanning 21 months of an organization's email server log, for 628 individuals. Emails to and from extra-organizational entities were removed, as were all "broadcast" emails. The server log data for each message included a message ID, the identities of the sender and recipients, and the time at which it was sent. For each message, the sender was removed from the set of recipients, if it was present. We also had access to the organizational hierarchy for 378 members of the organization, so we could calculate both the depth of each individual in the hierarchy (their distance from the top) and the number of subordinates that they supervised. We did not have access to the content or the message headers; thus, we knew neither how much (original) information a message m_i contained, its similarity or relations to other messages that the participants may have sent or received, nor whether it was in fact a reply to an existing message.

For privacy reasons, we do not refer to specific individuals in this data set by name.

We tested various values of f ($\{0.001, 0.01, 0.1, 0.9\}$) for both models for α_i. For the second model we set H to 1 day, based on observations made in [11], and set G to 1 hour.

We performed three separate sets of experiments: measuring the relations between our algorithms' rank ordering and properties of the organizational hierarchy, comparing our algorithms to others on the basis of their fidelity to the organizational hierarchy, and measuring the sensitivity of our algorithms' performance to parameter values.

The experimental code was written in Java, using the JUNG [8] libraries for network representation and analysis; some of the post-analysis used MATLAB. A single model instance required approximately 8 minutes to process the messages on a dual 2.5 GHz Apple PowerMac with 2.5 GB RAM (1.5 GB of which was allocated to Java heap space).

4. RESULTS AND DISCUSSION

As previously observed, there exists no ground truth to which we can compare the results of our ranking algorithms to determine their correctness. This is particularly true for the transient rank measurements ($R_i(c)$), since network ranking algorithms are generally applied to static networks representing all data. For this reason, evaluation of our models focused primarily on the derived cumulative rank measurements defined earlier: S_i, S_o, and S_r.

We represented the organizational hierarchy as a tree in which A is a child of B iff A is supervised by B; we then defined the *depth* in the hierarchy to be the number of steps from the root of the tree (that is, the person in charge of the organization), A's *subordinates* as the individuals in the subtree rooted at A (not counting A itself), and A's *superordinates* as the individuals on the path from A to the root of the hierarchy.

We derived ranks for HITS (hub score and authority score), PageRank (with random restart probability α of 0.1), and weighted indegree and outdegree by applying them to the network where X is connected to Y if X has ever emailed Y. We defined the weight of an edge $< X, Y >$ to be the sum of the weights of each message from X to Y (normalized appropriately for HITS and PageRank); the weight of a message was in turn defined as (a) proportional to $\frac{1}{|P_i|}$ (for HITS and PageRank) or (b) 1 (for weighted in- or outdegree, i.e., each individual is ranked according to the number of messages sent or received).

The following analyses focus primarily on the HITS authority score, PageRank, indegree, and S_i and S_r (sum of incoming potential and sum of transient ranks) models for ranking. These "inflow"-based and direction-agnostic ranking methods generally performed much better in this context than their "outflow"-based counterparts, so for reasons of space

the "outflow"-based results are largely omitted.

4.1 Rank and organizational hierarchy

Figure 4 shows the results of plotting the measured rank (from 0 to 627) against the hierarchy depth for several different ranking algorithms, where the distribution of ranks are presented as a box-plot[1] for each depth and each panel illustrates the results for a different ranking method. We observe the following:

- All of the ranking methods show that rank as determined from email is strongly dependent on tree depth in the organizational hierarchy: individuals who are highly ranked in the email data tend to be near the root node of the organizational tree, and vice-versa.
- The HITS authority ranking is the only one where the median rank does not monotonically increase with tree depth (there are 2 reversals). All of the other ranking methods are monotone in this sense–indeed their ranks are all strongly correlated with each other, while the ranks of HITS authority are much less correlated.
- It should be noted that tree-depth is not necessarily by itself a good predictor of importance in an organization. For example, at depth 2 in the tree are individuals who are likely to be administrative assistants or advisors to the individual at the root node, but who have no subordinates. We can see the existence of such individuals as outliers in several of the ranking plots for depth 2. (Below we look at a more subtle measure of organizational importance, namely the number of subordinates for each individual in the tree).

Figure 5 shows the results of plotting the rank values against the number of subordinates (for those with ≥ 1 subordinates) for the same ranking algorithms. We found that using the log of the number of subordinates produced a more interpretable plot compared to use the number of subordinates directly (which resulted in a very skewed plot)—in addition, the correlation between rank values and log-subordinates (for different methods) was significantly higher than for subordinates directly.

We see in Figure 5 a clear dependence between rank values and log(number of subordinates), with the exception again of HITS (authority ranking). In fact the dependence is weakly linear, as the correlation coefficients indicate (hovering around 0.4 and 0.5 for the higher correlations). Again, the InDegree, PageRank, baseline, and reply models were all highly correlated with each other with correlation coefficients of 0.8 and above (not shown).

Figure 5 confirms the results using tree-depth earlier: ranks based on email traffic are strongly correlated with the number of subordinates, at least for this data set. This is not particularly surprising, but nonetheless is informative to see

[1]The box summarizes the distribution of a value: the horizontal lines show the lower quartile, median, and upper quartile values; the vertical lines show the extent of the data; outliers are plotted separately[10].

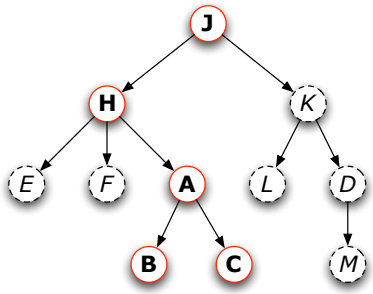


Figure 6: A sample hierarchy, with A's subordinates and superordinates highlighted.

borne out in practice. Outliers in these plots could for example be examined to see who they correspond to in the organization, e.g., employees who are low in the organization tree but who are ranked highly based on email traffic, or vice-versa.

4.2 Comparing rank algorithms

We hypothesize that an individual A should in general have higher rank than her subordinates and lower rank than her superordinates in the hierarchy tree. (We do not compare an individual to others than her sub- or superordinates, since it is not obvious that any consistent relationship ought to obtain between them.) Figure 6 highlights the subordinates (B, C) and superordinates (H, J) of A in a sample hierarchy tree.

On this basis, we say that, for a given algorithm, A is *inverted* with respect to its superordinate H if A's rank is higher than H's, and inverted with respect to its subordinate B if A's rank is lower than B's. We can then compare the performance of different ranking algorithms to one another based on measuring the inversions induced by each; an algorithm that corresponded perfectly to the hierarchy would have no inversions. Note that in general there are many different such rankings for a given tree; for instance, one can swap the ranks of sibling leaves (in Figure 6: B and C, or E and F) without affecting the number of inversions.

We can derive error measures from these inversions for each individual c in a few different ways: a simple summation of inversions, which we denote by $I(c)$; a weighted sum of inversions, based on rank difference or on depth difference (in which an inversion counts more if the rank/depth difference is greater), which we denote by $I_R(c)$ and $I_D(c)$ respectively; or a normalized count $I_N(c)$. I_N takes on values in $[0, 1]$, where 0 indicates no inversions and 1 indicates that all sub- and superordinates of c are inverted with respect to c); the additive inverse of this value is can be interpreted as an accuracy score. Note that I, I_R, and I_D place more emphasis on individuals with many subordinates, while I_N weights each individual equally.

Table 1 shows the result of calculating mean values for I, I_R, and I_N. (I_R and I_D turn out to be strongly correlated for this data set, so for simplicity we do not include figures for I_D here.) We observe the following:

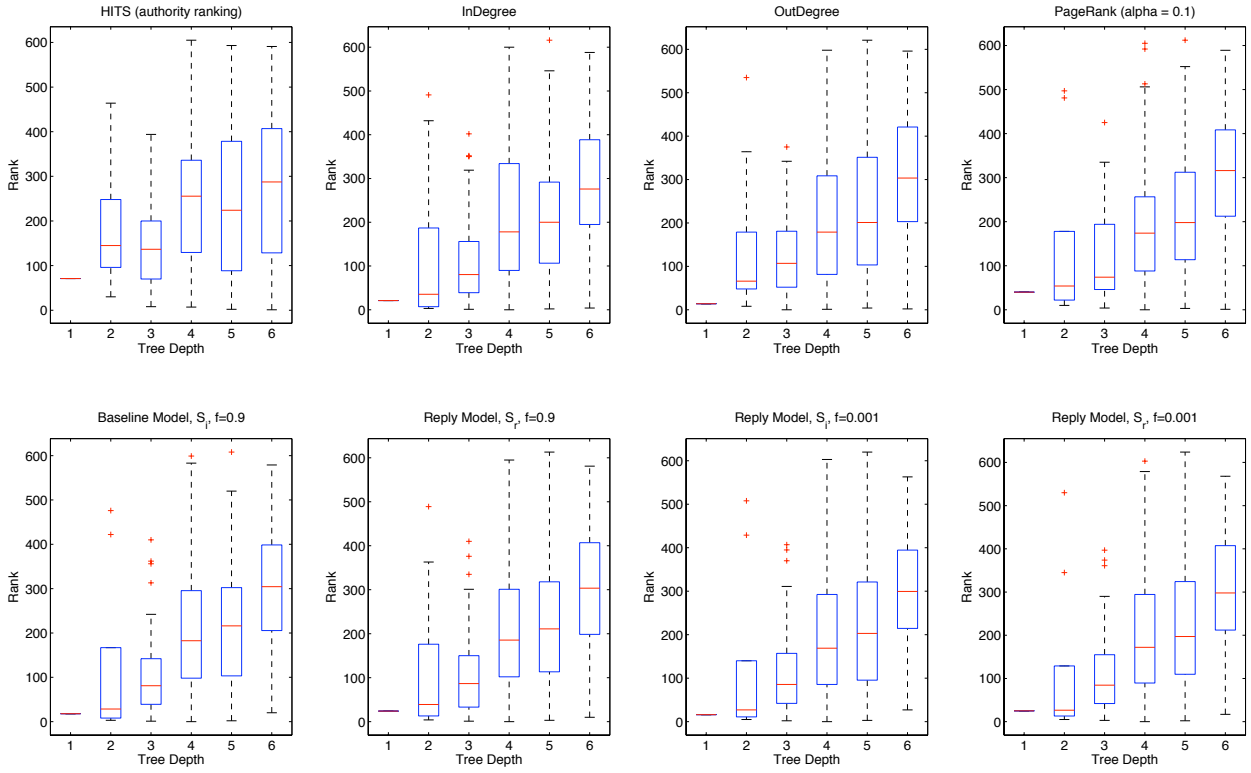


Figure 4: Rank versus depth for HITS (authority score), indegree, outdegree, PageRank, baseline model ($f = 0.9, S_i, S_r$), reply model ($f = 0.001, S_i, S_r$)

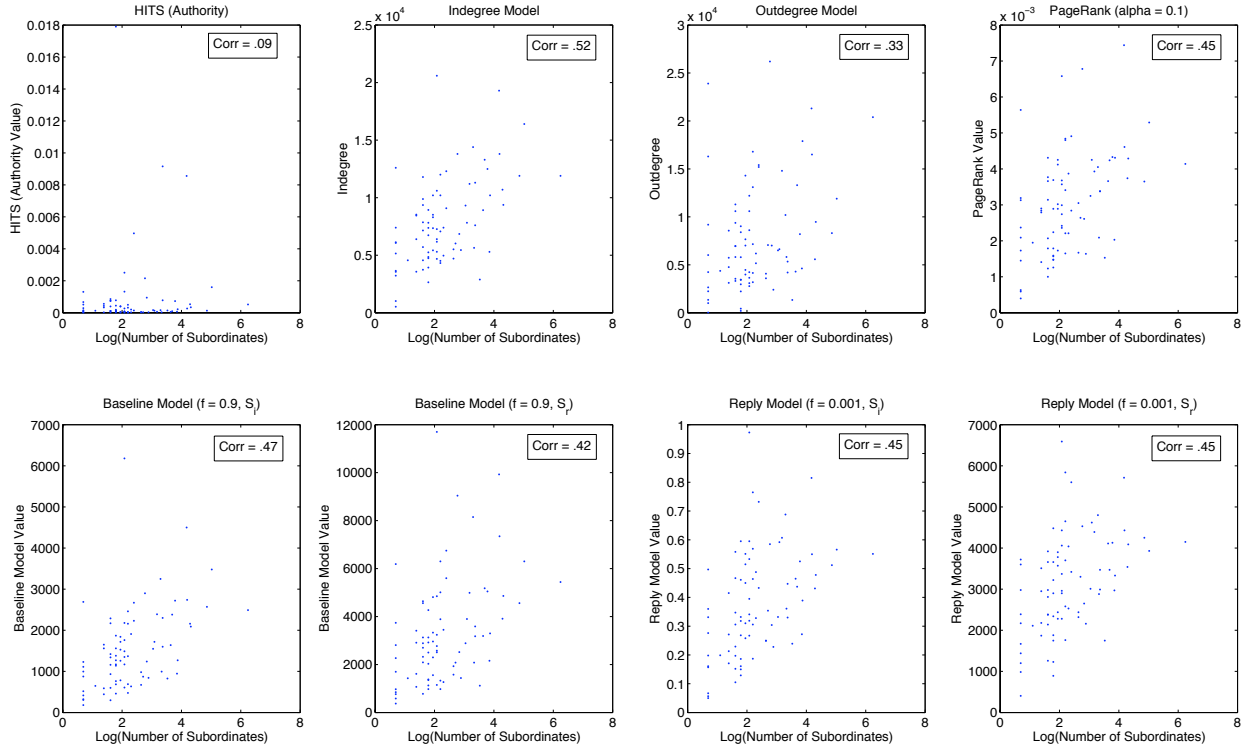


Figure 5: Rank value versus number of subordinates for HITS (authority score), indegree, outdegree, PageRank, baseline model ($f = 0.9, S_i, S_r$), reply model ($f = 0.001, S_i, S_r$)

	$\bar{I}$	$\bar{I}_R$	$\bar{I}_N$
HITS (authority)	1.17	48.88	0.88
PageRank ($\alpha = 0.1$)	0.80	41.14	0.92
indegree	0.54	39.38	0.95
baseline, S_i ($f = 0.9$)	0.41	16.05	0.96
baseline, S_r ($f = 0.9$)	0.63	33.33	0.94
reply, S_i ($f = 0.001$)	0.50	18.94	0.96
reply, S_r ($f = 0.001$)	0.55	30.15	0.95

Table 1: A comparison of the mean inversion scores of several ranking algorithms.

- S_i for the baseline and reply models outperforms all other ranking algorithms for all error measures.
- S_r for the baseline and reply models performs comparably to the indegree measure for I and I_N, and slighty outperform indegree for I_R.
- PageRank and the HITS authority score are outperformed by a significant margin. (The HITS hub score, not shown here, did rather worse than either.)

4.3 Sensitivity analysis

We evaluate the sensitivity of the rank orderings of models in this framework to the choice of input parameter values by building models using different parameter values, and then measuring the differences in rank orderings between each pair of models. We define the difference between two rank orderings as the mean absolute difference in rank ordering between each pair of individuals:

$$d(A_j, A_k) = \frac{\sum_{c \in C} |O_{A_j}(c) - O_{A_k}(c)|}{|C|} \tag{8}$$

where A_j and A_k are two different models, and $O_{A_j}(c)$ denotes the index of the rank assigned to c.

For a given entity set C, we observe that $d(A_j, A_k)$ takes on its maximum value when A_j produces an ordering that is the reverse of A_k's:

$$d_{\max}(A_j, A_k) = \frac{\sum_{k=1}^{|C|/2} (2k-1)}{|C|} = \frac{|C|}{2} \tag{9}$$

For this data set, therefore, the maximum mean difference is $628/2 = 314$; the figures below should be interpreted in the light of this information.

Figure 7 shows the result of cross-comparison of three ranking algorithms based on the basic model (S_i, S_o, and S_r), over variations in f. While varying f clearly has an effect on ranking–larger differences in f yield larger mean differences between the corresponding algorithms–the effect is small: the largest difference is $\approx$ 2% (6.02/314) for any pair of algorithms.

Figure 8 shows the result of cross-comparison of three ranking algorithms based on the reply model (S_i, S_o, and S_r), over variations in the parameters G and H, which are defined here to be functions of a single variable x: $G = 1800x$, and $H = 43200x$; larger values of x suggest an atmosphere in which email is generated and replied to at a slower pace. These figures indicate that varying x has a significant effect on rank ordering (again, larger differences in x yield larger mean differences in ordering): the largest mean difference in this case is $\approx$ 20% (65/314).

x	**0.5**	**1**	**2**	**4**
0.5	0.00	6.16	11.02	25.70
1	6.16	0.00	6.88	21.15
2	11.02	6.88	0.00	16.11
4	25.70	21.15	16.11	0.00

Table 2: Comparison of reply model algorithms ($f = 0.001$, varying x), for ranking method S_r.

The results shown in Figure 8 were generated using $f = 0.1$. We tested $f = 0.001$ as well, and found that the results for S_i and S_o (not shown here) were essentially identical, but that the results for S_r were markedly different; these results are shown in Table 2. We observe that a 100-fold reduction in f results in an approximately 2-fold reduction in mean difference magnitude for all values of x tested.

5. CONCLUSION AND FUTURE WORK

We have presented EventRank: a new framework, based on a set of clear requirements, for models that rank individuals in a social network derived from events occurring over time; these models respect event sequence and also provide a way of tracking rank changes over time as new events occur. Our experiments employed a novel method for evaluating the fitness of ranking algorithms when applied to a community with a known hierarchy, which involved evaluating the consistency of the rank ordering with the partial ordering specified by the organizational hierarchy.

Our preliminary investigation of this network, applied to an organizational email data set, has yielded promising results: our algorithms performed at least as well as the existing algorithms to which we compared them, and the orderings were shown to be a better fit with the organizational hierarchy.

Directions for future work include the following:

- application of these models to additional data sets, for further validation
- extension of the framework to incorporate header and content data; the reply model could be made more sophisticated, for example, if we knew which messages were in fact replies [7]
- application of this model to other types of event data, including undirected relations; the existing model does not depend on the fact that email events are directed
- investigation of methods for determining good values for f, G, and H based on requirements and time scales of interest
- analysis of transient rank values to automatically discover patterns in relative ranks of individuals over time, e.g., upward and downward trends in ranks for specific

f	**0.9**	**0.1**	**0.01**	**0.001**
0.9	0.00	4.44	5.79	6.02
0.1	4.44	0.00	1.81	2.12
0.01	5.79	1.81	0.00	0.44
0.001	6.02	2.12	0.44	0.00

f	**0.9**	**0.1**	**0.01**	**0.001**
0.9	0.00	3.72	4.69	4.87
0.1	3.72	0.00	1.30	1.64
0.01	4.69	1.30	0.00	0.50
0.001	4.87	1.64	0.50	0.00

f	**0.9**	**0.1**	**0.01**	**0.001**
0.9	0.00	2.18	3.12	3.37
0.1	2.18	0.00	1.26	1.58
0.01	3.12	1.26	0.00	0.42
0.001	3.37	1.58	0.42	0.00

Figure 7: Comparison of baseline model algorithms, varying f: S_i, S_o, and S_r respectively.

x	**0.5**	**1**	**2**	**4**
0.5	0.00	6.27	10.32	20.69
1	6.27	0.00	5.54	16.05
2	10.32	5.54	0.00	12.08
4	20.69	16.05	12.08	0.00

x	**0.5**	**1**	**2**	**4**
0.5	0.00	8.54	15.38	65.82
1	8.54	0.00	12.45	60.01
2	15.38	12.45	0.00	55.92
4	65.82	60.01	55.92	0.00

x	**0.5**	**1**	**2**	**4**
0.5	0.00	12.55	25.52	42.08
1	12.55	0.00	14.17	31.66
2	25.52	14.17	0.00	20.37
4	42.08	31.66	20.37	0.00

Figure 8: Comparison of reply model algorithms, $f = 0.1$, varying x: S_i, S_o, and S_r respectively.

individuals, periodic burstyness in ranks for individuals at certain times of year, etc.

6. ACKNOWLEDGMENTS

The authors wish to thank Danyel Fisher, as well as the reviewers, for their comments and feedback. This material is based upon work that was supported in part by the National Science Foundation under the Knowledge Discovery and Dissemination (KD-D) Program under Grant No. IIS-0083489.

7. REFERENCES

[1] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27:55–71, 2005.

[2] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[4] D. R. Gibson. Taking turns and talking ties: Networks and conversational interaction. *American Journal of Sociology*, May 2005. preprint.

[5] J. Heer. Exploring Enron: Visualizing ANLP results. *Applied Natural Language Processing*, 2004.

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

[7] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI*, 2005.

[8] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y.-B. Boey. Analysis and visualization of network data using JUNG. http://jung.sourceforge.net/doc/JUNG_journal.pdf.

[9] J. Seeley. The net of reciprocal influence: A problem in treating sociometric data. *Canadian Journal of Psychology*, 3:234–240, 1949.

[10] The Mathworks. MATLAB, 2005.

[11] J. R. Tyler and J. C. Tang. When can I expect an email response? A study of rhythms in email usage. In *Proceedings of the Eighth European Conference on Computer Supported Cooperative Work*, pages 239–258, September 2003.

[12] F. Wu and B. Huberman. Discovering communities in linear time: A physics approach. *Eur. Phys. Journal*, B38:331–338, 2004.

[13] P. S. Yu, X. Li, and B. Liu. On the temporal dimension of search. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 448–449, New York, NY, USA, 2004. ACM Press.

Discovering Important Nodes through Graph Entropy The Case of Enron Email Database

Jitesh Shetty
University of Southern California
University Park
Los Angeles, CA 90089
jshetty@usc.edu

Jafar Adibi
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
adibi@isi.edu

ABSTRACT

A major problem in social network analysis and link discovery is the discovery of hidden organizational structure and selection of interesting influential members based on low-level, incomplete and noisy evidence data. To address such a challenge, we exploit an information theoretic model that combines information theory with statistical techniques from area of text mining and natural language processing. The Entropy model identifies the most interesting and important nodes in a graph. We show how entropy models on graphs are relevant to study of information flow in an organization. We review the results of two different experiments which are based on entropy models. The first version of this model has been successfully tested and evaluated on the Enron email dataset.

Categories and Subject Descriptors

H.4 [**Link Discovery, Data Mining, Social Network Analysis**]: Miscellaneous; D.2.8 [**Graph Theory**]: Social Networks

General Terms

Graph theory

Keywords

Entropy, Link Discovery

1. INTRODUCTION

A new challenge in the area of Link Discovery (LD) [18]. and social network analysis (SNA) is to exploit communication pattern information and text information within knowledge discovery processes such as discovery of hidden organizational structure and selection of interesting prominent members. An interesting example of such a challenge is to discover hidden groups and prominent people by analyzing their email logs.

Email logs have been considered as a useful resource for research in such areas. Email logs are of prime importance and relevance in the study of information flow in an organization. Email has become the vital means of communication in the information community. Inherent advantages like ease of sending an electronic mail, archiving communications and the ability to reference past communications have made email the most acceptable and widely used means of communication. Though it is highly used in the business and professional domain its scope is not confined to it. Email is the most archived evidence data on interpersonal communication in electronic form. It can also act as an evidence database for law enforcement and intelligence organizations in their effort to detect hidden groups in an organization which are engaged in illegal activities. All these advantages make email a perfect test bed for relevant research like the study of information flow in an organization.

The study of information flow in an organization is germane to issues of productivity, efficiency and drawing some useful conclusion about the business processes of the organization. It can lead to insights on interaction patterns of employees within an organization at different levels of the organization hierarchy. Most of the experiments in this domain are performed on synthetic data due to lack of an adequate or real life benchmark. The recent availability of large datasets of human interaction like the Enron email dataset can be a touchstone for such research. This dataset shows intercommunication between employees of an organization hence it is perfect to study flow of information in an organization. This dataset is also similar to the kind of data collected for fraud detection or counter terrorism and hence it is a perfect test bed for testing effectives of techniques used for fraud detection and counter terrorism.

In this paper we adopt event based graph entropy (we refer to this as both ”event based graph entropy” or ”graph entropy”) to determine the most prominent yet interesting people in the Enron email dataset.

The rest of this paper is organized as follows. We begin with the problem of order in networks. Next, we describe our novel event based graph entropy model. At the end, we report our results of exploitation of such techniques on Enron dataset followed by related work and conclusion remarks.

2. ORDER IN NETWORKS

Most of the work in SNA nd LD represent their environment with a graph or network. We use both terms in this paper frequently. The question is what sort of mathematical model would work best. One way to describe a threat organization, or a social network is in terms of a graph. In this model, each node would represent an individual member and an edge linking two nodes would indicate direct communication between those two members. Mathematically, we may ask how many nodes must we remove from a given graph before it splits into two or more separate sub-graphs? For graphs of various sorts, it’s possible to estimate the probability that the removal of a certain number of nodes would split the graph into two or more separate units based on a set of policies and criterias. However, a

KDD ’2005 Chicago, Illinois

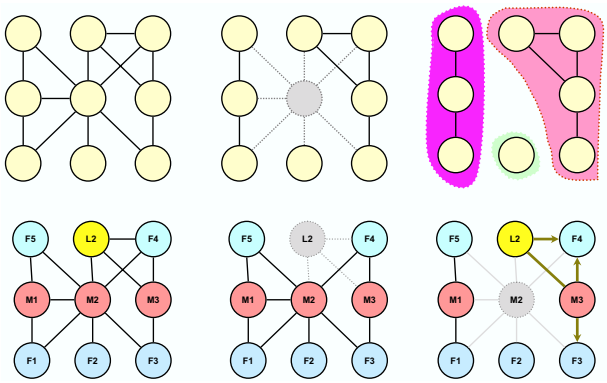


Figure 1: *Leaders* and *Followers* Example. L2 is *leader*, M1, M2 and M3 are *middlemen*; F1, F2, F3, F4 and F5 are *followers*. Up: example of a network. As it shows removing M2 splits the graph to three disconnected subgraphs Down: the same network after information about *leaders*,*middlemen* and *followers*. As it shows even though M2 splits the graph to three disconnected subgraphs, there are at least 2 pathes form *Leaders* to *followers* while removing L1 destroy such path.

graph model might not be the best representation of organizations such as drug dealers, terrorist organization and threat groups. In his recent work, Jonathan Farley explains clearly [6] that modelling terrorist networks as graphs does not give us enough information to deal with the threat. Modelling these networks as graphs ignores an important aspect of their structure, their *hierarchy*, and the fact that they are composed of *leaders* and *followers*. Hence, it is not enough to split the network since the remnant may contain a leader and enough followers to pursue their plans. [6] assume the network structure is known and authors try to find the optimum way to disrupt communication between leaders and followers. However, in our work we try to identify those important nodes as much as possible.

Figure 1 illustrates an example of such a phenomenon. The graph in the left shows a network consisting of three *leaders*: $L1$, $L2$ and $L3$; three *middlemen*: $M1$, $M2$ and $M3$; and three *followers* $F1$, $F2$ and $F3$. The graph in the right illustrates the same network without $M2$. As it is clearly seen that though such a removal splits the graph into two separate remnants, each sub-graph has leaders, middlemen and followers to carry orders and and execute the plan. Hence in this type of networks the relationship of one individual to another in a network becomes important. *Leaders* are represented by the topmost nodes in a diagram of the ordered set representing a network and *followers* are nodes at the bottom. Disrupting the organization would be equivalent to disrupting the chain of command, which allows orders to pass from *leaders* to *followers*.

Hence, the interesting problem here is to determine important nodes or leaders in a network. In other word, we are looking for those nodes whose removal has the maximum effect on the command chain.

3. GRAPH ENTROPY

We assume we have an evidence database (EDB) full of transactions among individual such as *email, Phone Call* etc. After exploiting the various explicit and implicit evidence fragments given in the EDB, we try to identify prominent members in a graph by looking at their transactions with others. To find prominent people in a network, we need to aggregate links between them and discover which node has the most effect on such a network. The entropy model can identify an entity or a set of entities which has the most effect on the graph entropy and thus provide a ranked list based on such effect. To do this we need to exploit facts such as individuals sharing the same property (e.g., having the same address) or transactions like being involved in the same action (e.g., sending email). Since such information is usually recorded by an observer we refer to it as *evidence*. Without loss of generality we only focus on individuals' actions in this paper, but not on their properties.

We transform the problem space into a multigraph $G =< V, E >$ in which each node represents an entity (such as a person or organization) and each link (edge) between two entities represents an action they are involved in. The term multigraph refers to a graph in which multiple edges between nodes are either permitted. For abstraction we summarize the set of actions (e.g., emails, phone calls etc in each edge and refer them as *link*) . Hence each *link* represent a set of actions in a vector. For instance an edge e_7 could be a set of two actions as $e_7 = [a_2, a_5]$. Also please note that it's possible to distinguish between email sender and receiver.

$V = \{v_1, \ldots, v_{|V|}\}$ Number of vertices
$E = \{e_1, \ldots, e_{|E|}\}$ Number of edges
$A = \{a_1, \ldots, a_{|A|}\}$ Type of actions

The EDB consist of tables representing individuals and actions among them at a given time. The table in Figure 2 shows an example of such data.

Assume we have a small society of 4 people who have been in contact with each other through actions. Figure 2 shows an example of such a database. There are four people and three possible actions: sending *Email ,making* a*Phone Call* and participating in a*Meeting*. When a person is not involved in any of the abovementioned actions at a particular time we show with action φ.

Hence $V = \{v_1, v_2, v_3, v_4\}$, $E = \{e_1, e_2, e_3 e_4\}$ and $A = \{Email, phoneCall, Meeting, \varphi\}$. For the matter of representation we show A as $A = \{E, C, M, \varphi\}$. The table in 2 illustrates actions among these individuals along with the action time.

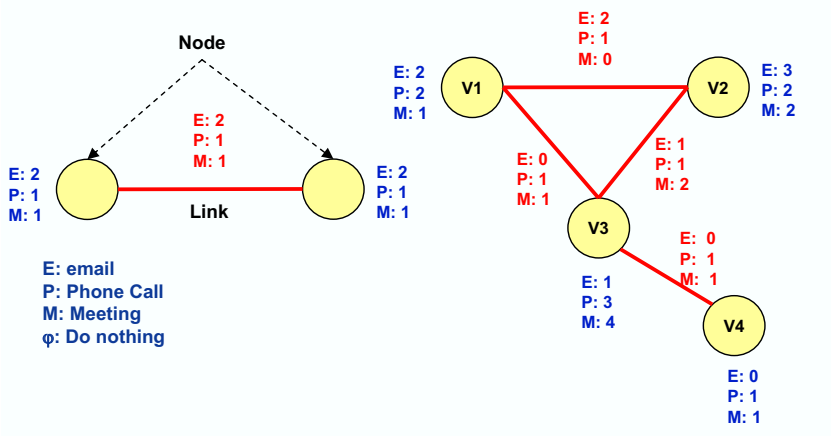
This graph has a major conceptual difference with well-known Bayesian and other similar graphical representations. Unlike such conventional techniques in which nodes are variables and links are statistical relation among variables (causal relations), here nodes represent entities, and links are relations among entities.

3.1 Graph Entropy

There is no commonly used definition of graph entropy. Indeed, one can define the graph entropy as the Kolmogorov complexity of its adjacency matrix and one can even use this definition to obtain interesting theoretical bounds for several important graph characteristics , but the Kolmogorov complexity is incomputable [4].

In the following we adopt the notion of graph entropy which is equal to Korner definition [13]of graph entropy when the graph is complete. Korner definition of graph entropy also has this limitation that all elements of the graph are being emitted by a discrete memoryless and stationary information source according to the probability distribution P. We show how we add memory to graph entropy definition by looking at sequences with length greater than 1.

Korner gave several descriptions of graph entropy $H(G, p)$ including the following.



Time	Action	V1	V2	V3	V4
1	V1 Email V2	E	E	Φ	Φ
2	V3 Phone Call V4	Φ	Φ	C	C
3	V2 Meeting V3	Φ	M	M	Φ
4	V1 Email V2	E	E	Φ	Φ
5	V2 Email V3	Φ	E	E	Φ
6	V1 Phone Call V3	C	Φ	C	Φ
7	V1 Phone Call V2	C	C	Φ	Φ
8	V2 Meeting V3	Φ	M	M	Φ
9	V3 Meeting V4	Φ	Φ	M	M
10	V1 Meeting V3	M	Φ	M	Φ
11	V2 Phone Call V3	Φ	C	C	Φ

Figure 2: MI Example. V1, V2, V3 and V4 represent people. *E, C, M*and φ stand for *Email, phoneCall, Meeting* and *doing nothing* respectively. The table on the right shows activities among people and the graph on the left illustrates such network.

$$H(G,P) = min_{x \in StableSet(G)} \sum_{v \in V(G)} p_v log(p_v) \qquad (1)$$

Where $StableSet(G)$ denotes the family of stable sets in vertices of G. A subset of vertex set is called stable set if it does not contain any edge. Stable sets in graphs form one of the important models in integer programming and have various applications. However, the stable set problem is NP-hard and also not easy to treat in practice. Even though there are some approximate ways to calculate such a set our definition of graph entropy is a special case of such a definition. However we extend such a definition to cover dependencies in the graph.

Let $G =< V, E >$ be a graph. Let P be the probability distribution on the vertex set $V(G)$. We will think of $V(G)$ as a finite alphabet. How we define such a alphabet depends on the nature of the problem. This definition has similarities with [16].

$$H(G,P) = \sum_{i=1}^{|V|} p(v_i) \log(1/p(v_i)) \qquad (2)$$

In general if we plot $H(x)$ in terms of $p(x)$ there are two sides of the curve that play an important roles. Those x with high probability and x with lower probabilities. We believe our model finds those instances. Figure 3 illustrates such phenomenon.

A great concern in LD domain is that elements of the data are not independent. For instance if the link $AsendemailtoB$ and link $BsendsemailtoC$ are dependent to each other, this means B may forward $A's$ email to C. Hence, we can change the probability space from $length = 1$ to $length = 2$ and more. This means our space consists of sequence of emails if the second one is dependent to the first one and so on.

Since discovering such dependency is not easy we provide three approaches to address such an issue. In the following we describe these cases.

- For every single transaction (for example *email*) we examine if it is similar to other received emails by a given individual. i.e. if she forwards an email, or copy and pastes a major part of an email.
- If a transaction happens immediately right after a given transaction (for instance if

  $$Time(transaction_i) - Time(transaction_j) < window$$

  we consider that a dependent transaction.
- Another alternative is exploitation of Markov Blanket type of model. In this model we assume an event (link) between two nodes is only dependent to those node's events (links connected to those nodes). For instance in Figure 3.1 we assume *red (dark)* event is only dependent to rest of the *black* links. In a more advanced model for any event e we can drive a set of dependent events such as $D_e = \{d_e^1, \cdots, d_e^{|D|}\}$ each with the probability of $P_e = \{p_e^1, \cdots, p_e^{|D|}\}$which shows the probability of dependencies to e. This probability could be derived from domain knowledge.

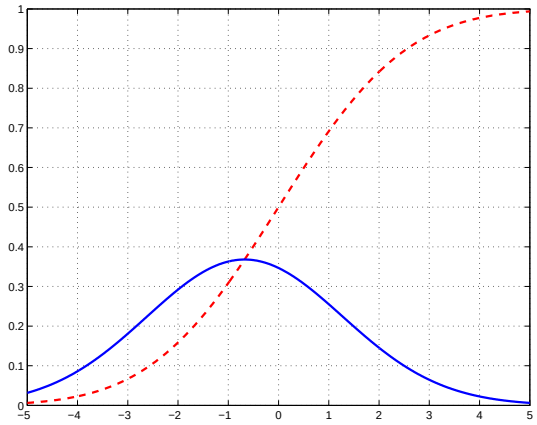


Figure 3: ($P(x)$ is a normal distribution, and the bell curve is the distribution of $H(x)$. We are mostly interested in right and left part of the $H(x)$.

We extend this notion to cover deeper levels of dependencies. For example, consider the domain of emails. A first level measure of graph entropy would be the predictability of an arbitrary email within that graph. In this approach, X would be the set

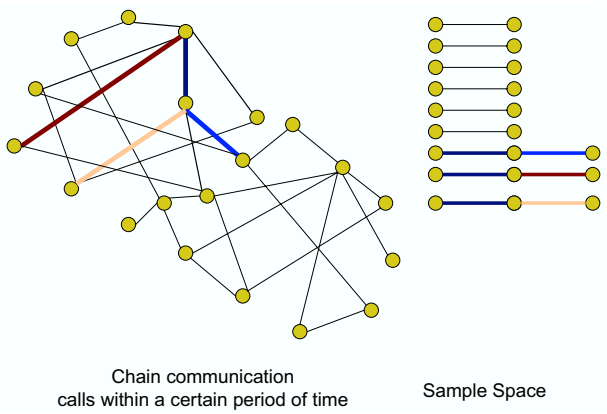


Figure 4: Dependent links

of all emails such as $A_{email}B$ contained in the graph. Furthermore, $P(A_{email}B)$ as the number of occurrences of $A_{email}B$ in the graph, divided by the size of the graph.

A more sophisticated approach would be to let X be all substrings of a certain length n. $P(X)$ would then be the number of occurrences of sequence X, divided by the total number of possible sequences with length n in the graph. As an example, let us choose length of $n = 2$. Hence we are counting sequences such as $A_{email}B_{email}C$ and $B_{email}D_{email}E$ and $p(A_{email}B_{email}C)$ would be the number of occurrences of such sequence over all possible sequences with length equal to 2 in the graph.

There are couple of issues associated with this definition. First of all it is obvious from entropy definition that more the regularity in sequence of events, the lower its graph entropy will be. As certain sequences occur more frequently, the probabilities of these sequences will increase, and probabilities for other sequences will decrease subsequently. As we mentioned earlier entropy is highest when the probabilities are uniform, and it decreases as the probability distribution becomes less uniform. Second, based on our definition there is no single entropy measure for a given graph; the value is dependent on the selected alphabet size, n. The value of n depends on the nature of the database and comes form intuition and domain knowledge. $n = 1$ only measure the entropy of nodes labels, without considering relationships between individuals. On the other hand very large number of n make the whole calculation very expensive and the interpretation will be very difficult.Finally, we consider the time of an event when we make our alphabet. Hence if $Time(A_{email}B) > Time(B_{email}C)$ we do not consider $A_{email}B_{email}C$ as a sequence.

3.2 Important Nodes

Our interpretation of important nodes are those who have the most effect of the graph entropy when they are removed from the graph. The intuition for this idea is that those who send more commands through the network and their messages are forwarded are important. In addition those who send unusual messages through the network also might be important people. To do this we execute the following procedure. First we calculate the entropy of the whole graph. Next for all nodes in the graph we remove them one by one and recalculate the graph entropy for the remnant graph. Following table illustrates such procedure.

Pseudo Code for Discovering Important Nodes

1. Compute the graph entropy using 2 as $Entropy_all$
2. For all nodes $N(i)$ in the do the following
3.
   - Compute the entropy of one node $N(i)$ by calculating the entropy of all of its edges as $E(i)$
   - Drop $N(i)$ from the graph
   - Calculate the entropy or remnant graph as $EN(i)$
   - Calculate the cross entropy of $EN(i)$ and $E(i)$
   - $Effect(i) = EN(i)/log(EN(i)/E(i))$
4. Rank nodes based on $Effect(i)$

4. EXPERIMENTAL RESULT

Below we report the results of applying the graph entropy model to the Enron Email Dataset [1]. There are many reason for using Enron dataset to evaluate our techniques. First of all, it is probably the only actual corporate email dataset available to public. Second, email logs are of prime importance and relevance in the study of information flow in an organization. Third, the study of information flow in an organization is germane to issues of productivity, efficiency and drawing some useful conclusion about the business processes of the organization. Finally this dataset is also similar to the kind of data collected for fraud detection or counter terrorism and hence it is a perfect test bed for testing effective of techniques used for fraud detection and counter terrorism.

The Enron email dataset was made public by the Federal Energy Regulatory Commission during its investigation. Database was later collected and prepared by Melinda Gervasio at SRI for the CALO (A Cognitive Assistant that Learns and Organizes) project; most of the integrity problems in the dataset had been resolved. It contains all kind of emails personal and official. Some of the emails have been deleted as part of the redaction effort due to requests from affected employees. William Cohen from CMU has put up the dataset on the web for researchers [2]. This version of the dataset contains around 517,431 emails from 151 users distributed in 3500 folders. The dataset contains the folder information for each of the 151 employees. Each message present in the folders contains the senders and the receiver email address, date and time, subject, body, text and some other email specific technical details.

We created a *MySql* database [3] for the dataset to catalyze the statistical analysis of the data and cleaned the dataset by removing a large number of duplicate emails. Folders such as *discussion threads* and *all documents* were generated by the computer and were not user created. We cleaned up the whole data to make it ready for our purposes. For detail of the data cleaning please refer to [19]. Our cleaned Enron email dataset contains 252,759 messages from 151 employees distributed in around 3000 user defined folders. Our prototype is written in Java and visualization made either by in-house developed Java Applet or using NetDraw [2]. The prototype is applicable to apply to any similar dataset. The database

[1]This database contains private emails, while reading this paper please be considerate about the privacy of the people who were not involved in any of the actions which precipitated the investigation. Authors do not attach any label to anyone in this dataset by no means. The main purpose of this study is to evaluate some novel techniques on actual real world dataset.

[2]http://www-2.cs.cmu.edu/ enron

[3]http://www.isi.edu/ adibi/Enron/Enron.htm

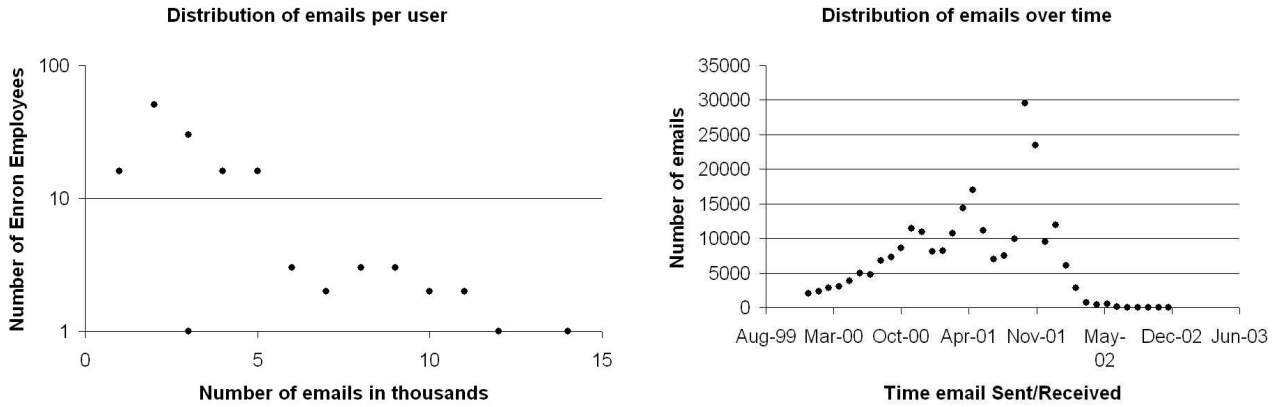


Figure 5: Enron Database distribution. Left: distribution of the messages per user. Right: distribution of the emails over time

scheme is very intuitive and general which make it easy to map to any other email dataset. A report on Enron database schema and dataset characteristics is available at [19].

Figure 5 (left) shows the distribution of the messages per user. The x-axis represents the number of email messages in log scale. The y-axis represents the number of Enron employees in log scale. The graph clearly shows that the messages are not evenly distributed between the users. A small number of users have a large number of messages. However, there are employees distributed through out the y-axis which reflects that the dataset contains employees with all amount of email messages. Figure 5 (right) shows the distribution of the emails over time. The figure clearly reflects that most of the emails have been sent and received in the year 2001. The x-axis represents the year in which the email has been sent or received and the y-axis shows the number of emails.

To illustrate the Enron network, we transform the Enron database into a graph as we discussed; each vertex of this graph represents an Enron employee. An edge exists between two employees if the two employees have exchanged emails. This graph constitutes of 151 employees of Enron. The graph is shown in Figure 6. We found out the position of every employee in the ex organization hierarchy. The color of the nodes stands for the position of the employee in the ex organization. The major type of communication are "TO" and "CC".

The Enron email database has more than 70K emails which are referenced emails; these are emails which refer some other emails. But another scenario is where a particular email doesn't technically refer some other email but has relevant information. This bring up a very interesting phenomenon with the original Enron graph, the edges which represent exchange of information don't end at the receiver node but the information flows much deeper into the Enron graph involving a lot of other nodes. Here we expand the scope of influence of nodes to every such node which share a particular information. There are certain intricate issues involved in detecting referenced emails and in particular detecting a pattern of how some particular information was conveyed to other nodes in the graph. There is no evidence in the database about the generator node or transient nodes of the forwarded emails. Also when some information is conveyed further there is some more information added or the original information changed. We detect the referenced emails based on the percentage similarity with the original email.

4.1 Enron Important Nodes

We compute the entropy of the entire Enron graph. We then drop a node and also drop the edges fanning in and out that particular node and recalculate the entropy. We measure the change in graph entropy. We do this for each node present in the graph. We generate a ranked list based on the change in graph entropy. We conclude that the node whose absence brings maximum change in graph entropy is the most influential node in the graph.

We repeated such procedure for the following two experiments.

1. Sequence of length = 1. Here we only consider emails among individuals as our space. This is the procedure for detecting influential nodes in the graph using the entropy model at *length = 1*. The model at *length = 1* limits the scope of influence of every node in the graph to directly connected nodes. But past work using epidemic models [1] on social networks show that information is passed by hosts in a social network to other interested people in the network. This shows that when certain people are engaged in some activity in a network they pass information amongst each other and might not be in direct contact with each other.

2. Sequence of length = 2. In a network like the Enron graph there is a possibility that information might be hopped through nodes deliberately. This expands the scope of influence of nodes over other nodes. In the next step we calculate the graph entropy at *length = 2*. If a node in the graph is not directly in contact with some other node, but receives information from it through a third node, then its presence in the graph has influence over though they are not directly in contact with each other. This influence is taken into consideration in the *length = 2* computation of entropy.

To measure if an email is dependent to one of the the previous emails in a mail box we conducted the following procedure.

We created an Enron dictionary which contains all the words in the organization vocabulary, there are certain words which are not there in traditional dictionaries like organization jargons, some proper nouns etc. These words in the dictionary don't contain stop words and are stemmed words. Stop words are those words like conjunctions, prepositions and articles which do appear often in the document yet alone carry little meaning. We used the porter stemming algorithm for stemming the words. The porter stemming algorithm is a process of removing the commoner morphological and in flexional endings from the words in English. Its main purpose is as part of the term normalization process that is used when stepping up information retrieval systems. We normalized all emails using this. We generate a vector representation for each email. Then we compare the vectors using the Jaccards Algorithm.

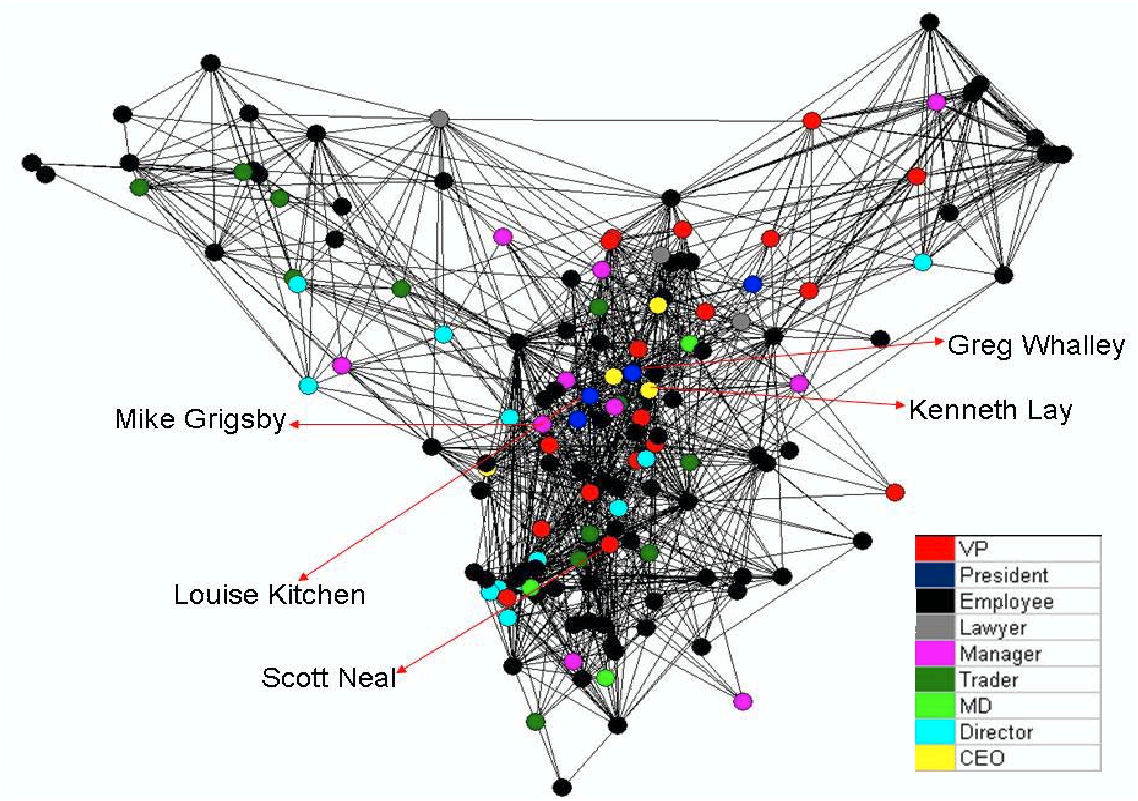


Figure 6: Enron Network

$$Similarity(X,Y) = \frac{X \bigcap Y}{X \bigcup Y} \quad (3)$$

The percentage similarity of the Jaccards algorithm is ratio of intersection of two vectors and their union. Thus the emails referenced by the original emails are all those emails which have more than 60% Jaccard score. We take the threshold as 60% based on emperical results. We performed experiments on the referenced emails in the Enron database and calculated their percentage text similarity. The average percentage text similarity between referenced emails in the Enron database is 63.71%. So we conclude that if any two emails are more than 60% similar the context of talk is same, and is thus linked.

The emails are further ordered based on the time stamp. This gives a hierarchy for each email if it has been referenced and the nodes at each hierarchy can be obtained from the database. Thus we can relate the influence of nodes to those edges which are not in direct contact. This is used in computing the entropy at *level 2*. In *level 2* computation we calculate the entropy of the entire graph in the same way as we did for *level 1*. But then when we drop each node, we not only drop the edges spanning in and out from this node but we also drop those edges which have used this node in its path of information flow. So we drop edges which are not directly in contact with this node but they have either been originally generated from this node or used this node as a transient, and calculate the change in entropy. Then we generate a ranked list/seed group based on the change in graph entropy. The results of this for the Enron graph are shown in next section.

The results for *length = 1* are shown in Table 1. *Louise Kitchen* the ex president of Enron online is the most influential node in the Enron graph based on the entropy model at *length = 1*. The second most influential node in the Enron graph is *Mike Grigsby* who is an ex manager at Enron, followed by *Greg Whalley* ex president, *Scott Neal* ex employee and *Kenneth Lay* ex CEO. We now generate the ranked list/seed group based on the entropy model at *length = 2* as discussed earlier. At *length = 2* we take into consideration the information which has not been received from a direct contact but the information has been forwarded from some other node in the graph. Here again we show only the first five members in our ranked list/seed group. The results for *length = 2* is illustrated in 2.

Table 1: Most Important Nodes $length = 1$

Rank	Name	Designation at Enron
1	Louise Kitchen	President
2	Mike Grigsby	Manager
3	Greg Whalley	President
4	Scott Neal	Employee
5	Kenneth Lay	CEO

In the ranked list group generated based on the entropy model for *level 2* members like *Greg Whalley* and *Kenneth Lay* have a higher rank. *Louise Kitchen* and *Mike Grigsby* get lower ranks. The *length = 2* computation expands the scope of influence of each node over other nodes in the graph.

Table 4 compares graph entropy model result with some other conventional techniques such as betweenness centrality. Clearly betweenness centrality capture those nodes that are in the center of the graph but not necessarily those with higher authorities. Though since Louis Kitchen had a crucial role in Enron and several VPs and

Table 2: Most Important Nodes $length = 2$

Rank	Name	Designation at Enron
1	Greg Whalley	President
2	Kenneth Lay	CEO
3	Louise Kitchen	President
4	Mike Grigsby	Manager
5	Harry Arora	VP

Table 3: List of people with high number of sent emails

Rank	Name	Designation at Enron
1	Jeff Dasovich	N/A
2	Kay Mann	Employee
3	Sara Shackleton	N/A
4	Tana Jones	N/A
5	Chris Germany	N/A

Managers used to report to her she is in the betweenness centrality list as well.

In addition, we compare the result of our model with a simple frequency counting of those individuals who have sent most emails comparing to the rest of the Enron employees. Table 3 illustreates these people.

5. RELATED WORK

As illustrated by our experiments the main focus of our work is to find important nodes in a graph. We further use this to find relations and connections among entities and individuals. Our approach does not look for similarities among individuals as a classification task such as the work by Getoor et. al [8].

Graph entropy has different definition in various literature depending on the nature of the data [13] [11] [14]. We use a different notion of graph entropy and consider dependencies among links as well. Similar notion of such definition is introduced in [16].

In his recent work [3] Borgatti address the problem of discovering *key players* in a network. His approach is based on measuring explicitly the contribution of a set of actors to the cohesion of a network. In addition, he identifies two separate conceptions or functions of key players which reflect different analytical goals, and develop separate measures of suitability for each type of goal. In addition our approach has a fundamental difference with [3]. While Borgatti finds key players in a network, we try to find leaders. Our example at the beginning of this paper illustrates that key players are different with influential nodes. Freeman [7] in his work address centrality issues in social networks. As we discussed earlier the concept of centrality is close to key players and it is different with our view and definition of important nodes.

In [20] they address the problem of most important nodes in the network. One major difference of this work with our work is that we do not consider the Google referral type of links. Their example of bibliographical is based on reference which make the problem somewhat different. Famous works such as Google Page Rank [17] and HITS [12] are also in this category.

In [5] a linear model is used based on well-known electrical circuits formulas to represent a graph. They produce approximate, but high-quality connection subgraphs in real time on very large graphs. [9] also uses the same approach and exploits Kirchhoff laws to model the social network graph. Other approaches such as [15] which use betweenness and centrality to find crucial central nodes.

Table 4: Most Important Nodes Betweenness Centrality

Rank	Name	Designation at Enron
1	Bill Williams	Broker
2	Steven Merris	N/A
3	Eric Linder	Employee
4	Kay Mann	Employee
5	Louise Kitchen	President

Another issue is that event based graph entropy is scalable. We do not need to explore more than 3 or 4 levels to find important nodes. We can explore the graph around those nodes and run the engine recursively to find more important nodes. Approaches similar to betweenness centrality are also effective but may fail when applied to large networks, since the order of the algorithm is at least N^2 where N is the number of nodes.

Scale-free network also has been discussed extensively recently in literatures. One of the major line of work in scale free networks is gossip modelling and finding the *most influential nodes* to either broadcast a gossip or to prevent a virus distribution over a given scale free network. Kempe et al in their work [10] they consider the problem of selecting influential nodes. Using an analysis framework based on submodular functions, they show that a natural greedy strategy obtains a solution that is provably within 63% of optimal for several classes of models. Our work differentiates from this work since they do not have the notion of order in their model and their definition of *most influential nodes* is somewhat different with our *definition of important nodes*.

Our work is inspired by [6] in which he introduces the notion of order in networks and graphs. However we used a different approach comparing to [6].

6. CONCLUSION

The Enron email dataset is the largest real email dataset present in the public domain; other datasets haven't been public because of privacy concerns. This dataset contains all kind of emails personal and official. This is a valuable resource for many diverse fields. Social network analysis, link discovery are the most relevant fields where this can be used.

In this work, we defined and addressed the problem of important nodes and finding closed group around them. Additional contributions are the following:

- We proposed a novel yet simple intuitive way to measure the graph entropy as event based graph entropy. We showed that approaches like betweenness centrality lead to poor answers when our network consist leaders and followers.
- We provided a systematic way to find important nodes in a graph based on their effect on graph entropy.
- Moreover, we implemented our algorithms in a working prototype, complete with an interactive Java-based interface, on a real graph that we derived from the Enron Dataset. The graph has about 150 nodes and more then quarter million links.

In this paper we focused on using event based entropy to find influential nodes in a graph. We tested an evaluated the results on the Enron graph. Enron graph being a representation of a real life ex organization there was some evidence available to validate some facts revealed from our experiments. There are certain basic assumptions on which the entropy model claims its results, like the

evidence data is complete and there is no noise in the data. The result gets deteriorated when used on noisy data. Our main focus was to exhibit how an entropy model can act as a good means for detecting influential nodes in a graph.

There are several lines of ongoing and future work, such as, determining group leaders by measuring their entropy over time to capture the change in such entropy in a given period. In addition we would like to exploit sampling, randomization and data streams techniques to deal with very large datasets.

7. ACKNOWLEDGMENTS

This work was supported in part by the Department of the Navy, Office of Naval Research under contract N00173-05-1-G006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research. The authors would like to thank Hans Chalupsky and Eduard Hovy for their invaluable comments.

8. REFERENCES

[1] L. Adamic and B. Huberman. *Information dynamics in a networked world*. Lecture Notes in Physics. Springer, 2003.

[2] S. Borgatti and R. Chase. *Net Draw*. http://www.analytictech.com/, 2004.

[3] S. Borgatti. Identifying sets of key players in a network. In *Computational, Mathematical and Organizational Theory.*, 2005.

[4] H. Buhrman, M. Li, J. Tromp, and P. Vitnyi. Kolmogorov random graphs and the incompressibility method. *SIAM Journal on Computing*, 29(2):590–599, 2000.

[5] C. Faloutsos, K. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004.

[6] J. D. Farley. Breaking al qaeda cells: A mathematical analysis of counterterrorism operations (a guide for risk assessment and decision making). *Studies in Conflict & Terrorism*, 26:399411, 2003.

[7] L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[8] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *IJCAI01 Workshop on Text Learning: Beyond Supervision*, Seattle, Washington, 2001.

[9] B. Huberman and W. Fang. Discovering communities in linear time: a physics approach. In *KDD*. ACM, 2004.

[10] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence in a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[11] J. Kieffer and E. Yang. Ergodic behavior of graph entropy. *ERA American Mathematical Society*, 3(1):11–16, 1997.

[12] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46, number =, 1999.

[13] J. Korner. Bounds and information theory. *SIAM Journal on Algorithms and Discrete Mathematics*, (7):560–570, 1986.

[14] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 1997.

[15] M. Newman. Who is the best connected scientist? a study of scientific coauthorship networks. In E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, *Complex Networks*, pages 337–370. Springer, 2004.

[16] C. Nobel and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. Technical report, Stanford, 1998.

[18] T. Senator. Evidence extraction and link discovery. 2002.

[19] J. Shetty and J. Adibi. Enron email dataset. Technical report, USC Information Sciences Institute, http://www.isi.edu/ adibi/Enron/Enron.htm, 2004.

[20] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275, 2003.

GiveALink: Mining a Semantic Network of Bookmarks for Web Search and Recommendation

Lubomira Stoilova
Comp. Sci. Dept.
Indiana University
lstoilov@cs.indiana.edu

Todd Holloway
Comp. Sci. Dept.
Indiana University
tohollow@cs.indiana.edu

Ben Markines
Comp. Sci. Dept.
Indiana University
bmarkine@cs.indiana.edu

Ana G. Maguitman
School of Informatics
Indiana University
anmaguit@cs.indiana.edu

Filippo Menczer
Informatics & Comp. Sci.
Indiana University
fil@indiana.edu

ABSTRACT

GiveALink is a public site where users donate their bookmarks to the Web community. Bookmarks are analyzed to build a new generation of Web mining techniques and new ways to search, recommend, surf, personalize and visualize the Web. We present a semantic similarity measure for URLs that takes advantage both of the hierarchical structure of the bookmark files of individual users, and of collaborative filtering across users. We analyze the social bookmark network induced by the similarity measure. A search and recommendation system is built from a number of ranking algorithms based on prestige, generality, and novelty measures extracted from the similarity data.

Keywords

Semantic Similarity, Collaborative Filtering, Web Search, Social Bookmark Networks

1. INTRODUCTION

The GiveALink project is an attempt to explore alternatives to centralized search algorithms. Traditional search engines today crawl the Web in order to populate their database. When a user submits a query, results are generated and ranked using text similarity measures, the hyperlink structure of the Web, and click-through data from the company's servers.

GiveALink distributes the process of collecting data and determining similarity relations among all of its users. We use bookmark files as a convenient existing source of knowledge about what Web pages are important to people, and about the semantic structure in which they are organized. All of the URLs in our database originate from bookmark files donated by users. We further determine similarity relationships and relevance to queries by mining the structure and the attribute information contained in these files. Thus we propose a notion of similarity that is very different from the ones used by Google, Yahoo and MSN. Our measure of similarity is not based on the content of the pages in our database and not even on the Web link graph. Instead, it is an aggregate of the independent notions of semantic similarity contributed by different bookmark file owners.

There are several other Web sites that collect bookmarks and provide services such as sharing, tagging, and full-text search. These include `Del.icio.us`, `Simpy`, `Furl`, `Spurl`, `Backflip`, `CiteULike`, and `Connotea` and are reviewed by Hammond et al. [6]. GiveALink is different in that we actively exploit both collaborative filtering and the hierarchical structure of bookmark files, where present. We develop novel Web mining techniques and applications that are not already available elsewhere. Furthermore, GiveALink is a non-commercial research project and both our data and algorithms are openly available to the Web community.

LinkKDD'05, August 21, 2005, Chicago, IL, USA

2. BACKGROUND

The GiveALink system collects donated bookmark files and applies collaborative filtering techniques to them to estimate the semantic similarity between the bookmarked URLs. This section introduces definitions and some previous work related to collaborative filtering and semantic similarity. It also discusses briefly the type of information contained in bookmark files.

2.1 Collaborative Filtering

Collaborative filtering, also referred to as social information filtering, identifies and exploits common patterns in the preferences of users. Traditionally, it has been used to identify communities and build recommendation systems based on like users opinions. Examples include Ringo [16] for personalized music recommendations and GroupLens [12] for filtering streams of net news, as well as e-commerce sites such as Amazon.com [17] that make personalized product recommendations. These systems are predicated on the assumption that individuals who have shared tastes in the past will continue to share tastes in the future.

Collaborative filtering techniques are also used for inferring global structures in information domains. A prominent example is Page-Rank [3], a global measure of citation importance for URLs. To a first-degree approximation, PageRank assumes that the number of citations or inlinks to a Web page is a testimony for its importance and quality. Of course, the hyperlink structure of the Web has been created by individual users adding links from their pages to other pages. Thus the count of citations to a Web page is in essence a collaborative filtering measure.

Despite their success and popularity, collaborative filtering techniques suffer from some well-known limitations [15]. One is the sparsity of user profiles: the number of items contained in one user

profile is negligible compared to the entire dataset (the Web in our case). Thus the information contributed by one user is small and it is hard to infer communities because there is little overlap between profiles. Another critical limitation is the complexity of collaborative filtering algorithms. The latency associated with processing large data sets requires that similarity information is pre-computed offline. When a user submits a new profile, the data is not integrated into the system until the next time the database is rebuilt. The user profile is not updated until then either. Finally, collaborative filtering systems cannot generate predictions about new items. Since similarity is estimated based on existing user profiles, the system cannot generate recommendations for URLs that the users are not already familiar with. Many of these limitations also apply to the system described here.

2.2 Semantic Similarity

Semantic similarity between Web sites is a term used to describe the degree of relatedness between the $meanings$ of the Web sites, as it is perceived by human subjects. Measures of semantic similarity based on taxonomies (trees) are well studied [5, 8]. Recently Maguitman *et al.* [9] have extended Lin's [8] information-theoretic measure to infer similarity from the structure of general ontologies, both hierarchical and non-hierarchical. The ODP[1] — a human-edited directory of the Web that classifies millions of pages into a topical ontology — can be used as a source of semantic similarity information between pairs of Web sites.

Search engine designers and researchers have made numerous attempts to automate the calculation of semantic similarity between Web pages through measures based on observable features, like content and hyperlinks. Studies conducted by Menczer and colleagues [9, 11] report, quite surprisingly, that measures relying heavily on content similarity (e.g. common words) are very poor predictors of semantic similarity. On the other hand, measures that only take into consideration link similarity (common forward and backward edges), or scale content similarity by link similarity, estimate semantic similarity with greater accuracy. Incidentally, neither content nor link similarity alone is a good approximation of semantic similarity, and they are also not strongly correlated with each other.

Thus the question of how to automate the semantic similarity measure for arbitrary pages remains open. Here we propose another measure that is based on combining the semantic similarity notions of a community of users through collaborative filtering techniques, and compare it with previously studied measures based on content, links, and the ODP.

2.3 Mining Bookmarks

An issue that we need to consider before looking at the GiveALink system is the type of information we can expect to find in bookmark files. Bookmarks are a convenient source of knowledge about the interests of Internet users. On one hand, they are human-edited taxonomies and we have well-established techniques for extracting semantic similarity information from them. Additionally, they are designed to be easily transferable between browsers and computers and that makes it easy for users to access them and upload them to our Web site. We have considered using other sources of information, like browsing history files, which arguably contain more data. They present some technical and privacy challenges and may be considered at a later time.

Bookmark files contain a mix of explicit and implicit knowledge. The following attributes are explicit in the bookmark file: (1) URLs, (2) titles, (3) the hierarchical structure of the file, and (4) the browser and platform. Additionally, some browsers provide the time when bookmarks are added and last accessed, as well as personalized title and description that users can edit themselves.

We also exploit some of the implicit knowledge contained in bookmarks by taking into consideration the way people generally use these files. McKenzie *et al.* [10] report that people maintain large, and possibly overwhelming, bookmark collections. The bookmarked URLs are usually highly revisited, but they are seldom deleted and often some of the bookmarks are stale links. Additionally, Abrams *et al.* [1] suggest that people use bookmarks for different and sometimes unrelated reasons: some users bookmark URLs for fast access; other users bookmark URLs with long names that they cannot remember; yet others use bookmarks as a way to share their informational space with the community. Thus it is important to not make strong assumptions about the way the bookmark files were built when mining information from them.

3. THE GIVEALINK SYSTEM

This section presents the architecture of the GiveALink donation system and database. It also describes how we mine the collected bookmark files to build a URL-to-URL matrix containing semantic similarity values.

3.1 System Architecture

Users can donate their bookmarks anonymously or as registered users at `givealink.org`. To protect the database from bots that pollute it with a large quantity of engineered bookmark files, we require users to pass a CAPTCHA test [18] when donating anonymously. In addition, we prevent multiple submissions of identical files (like default bookmark files) by checking the MD5 signature of every donated file.

When users register, they have to provide a valid email address. We query the host to make sure that the email address is valid, and then issue the user an activation code. To activate the account, the user has to send an email to a special email address and include their activation code in the subject of the email. We use relay information from the email to verify that the email is coming from the correct source. This registration process is proposed by Jakobsson and Menczer [7] as an alternative to the double-opt in protocol to avoid email cluster bomb DDoS attacks.

When users donate bookmarks at `givealink.org`, we use their user agents to determine which browser and platform they are using in order to parse the file correctly. Our set of parsers supports Internet Explorer, Netscape, Mozilla, Firefox, Safari, and Opera. The file formats are as follows: Netscape stores bookmarks as HTML, Safari uses XML, and Opera keeps a simple ASCII list of bookmarks with their corresponding folders preceding them. Internet Explorer (IE) requires the user to export their bookmarks to the Netscape format because IE stores bookmarks in folders with one URL per file. Furthermore, Mozilla and Firefox both use the Netscape method of storing the bookmarks.

The back-end of the system is anchored by a MySQL database server. The data stored in the database includes users, browser and platform data, the directory structure of the bookmark files, the URLs themselves, as well as some personalized information about the URLs like descriptions that users entered, the time the bookmark was created and last accessed.

3.2 Bookmark Similarity

The URLs in a bookmark file are organized in directories and subdirectories and thus have an underlying tree structure. We view the bookmarks submitted by one user as a tree rooted at her username. Then we combine all of the user trees into a single tree by

[1]Open Directory Project, `dmoz.org`

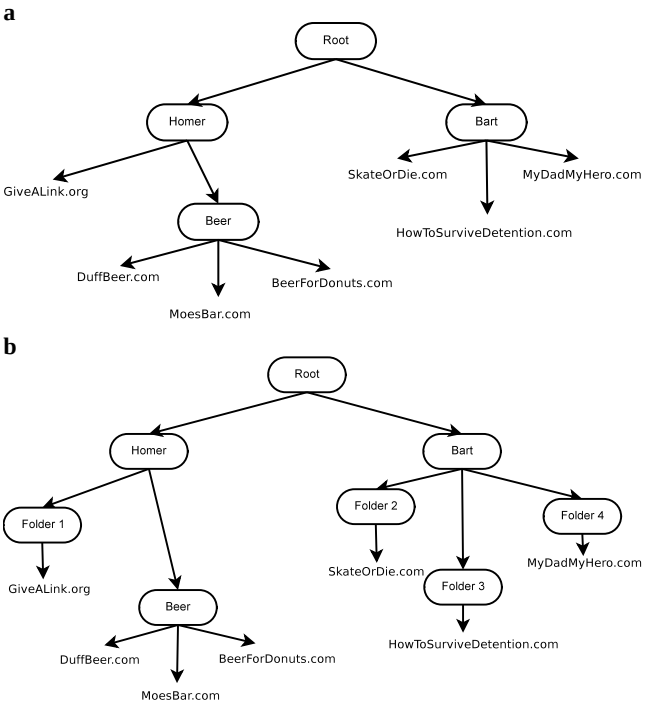


Figure 1: (a) An example tree containing the combined bookmark collection of two users, Homer and Bart. (b) The tree has been modified: each unclassified URL (i.e. each URL located in the user's root directory) has been given its own folder.

introducing a new root (super user) which is the parent of all user nodes. Figure 1(a) shows an example scenario in which only two users donated their bookmarks. Homer's bookmark collection contains `givealink.org` and also a folder called *Beer* that contains three URLs. Bart's collection is a flat file with three URLs. In our system, these URLs are stored as a single tree where the root is a parent of directories Homer and Bart, which are in turn the parents of the respective user's bookmarks.

To exploit the hierarchical structure of bookmark files, we use Lin's measure [8] to calculate similarity between the URLs in a user u's tree. Let URL x be in folder F_x^u, URL y be in folder F_y^u, and the lowest common ancestor of x and y be folder $F_{a(x,y)}^u$. Also, let the size of any folder F, $|F|$ be the number of URLs in that folder and all of its subfolders. The size of the root folder is $|U|$. Then the similarity between x and y according to user u is:

$$s_u(x,y) = \frac{2 \times \log\left(\frac{|F^u_{a(x,y)}|}{|U|}\right)}{\log\frac{|F^u_x|}{|U|} + \log\frac{|F^u_y|}{|U|}}. \quad (1)$$

This function produces similarity values between 0 and 1. For example, if two URLs appear in the same folder, then their similarity is 1 because $F_x = F_y = F_{a(x,y)}$. Also, all other things being equal, the similarity between x and y is higher when F_y is a subfolder of F_x, than when F_x and F_y are siblings.

A downfall of this approach is that many Web users do not organize their bookmarks in folders and subfolders and instead keep a flat list with their favorite links. In this case, according to Lin's measure, all of the links are in the same folder, so they must have similarity 1 to each other, but in reality we cannot conclude strong semantic similarity between URLs listed in such unorganized files. Therefore if a user decided to leave some URLs in their root directory, we think of each URL as if it were in its own folder. Figure 1(b) depicts how we modify the tree from Figure 1(a) before calculating semantic similarity. In the modified structure, URLs listed at the root have similarity slightly higher than 0 (as opposed to 1 in the original structure).

Table 1: Protocols of bookmarks.

Protocol	Donations	%
http	19679	97.2 %
https	246	1.2 %
feed	123	1 %
file	97	< 1 %
ftp	53	< 1 %
javascript	33	< 1 %
gopher	1	< 1 %
other	13	< 1 %

According to Equation 1, two URLs donated by different users have $s_u = 0$ because the least common ancestor is the root (super user). Thus Lin's measure is only appropriate for calculating the similarity of URL pairs according to a single user. To calculate the global similarity between URLs x and y, we average the similarities reported by each user:

$$s(x,y) = \frac{1}{N}\sum_{u=1}^{N} s_u(x,y). \quad (2)$$

If a user has both URLs x and y, then he reports $s_u(x,y)$ according to Equation 1, otherwise he reports $s_u(x,y) = 0$. If a user has URL x in multiple locations, we calculate $s_u(x,y)$ for all locations of x and report the highest value. The final similarity matrix represents a weighted undirected graph where the nodes are URLs and the weight of an edge is the similarity of the two connected URLs.

4. SIMILARITY NETWORK ANALYSIS

As of June 5, 2005, GiveALink has 113 users who have donated a total of 22,065 unique URLs. Based on this initial small number of donors, there is relatively little overlap of bookmarked URLs between different users. As a result the similarity matrix is very sparse. The protocols of the donated URLs are shown in Table 1.

Figure 2 shows the topology of the graph. The well defined clusters suggest that our semantic similarity measure is able to identify communities of Web pages that share a topic. One of the clusters, *News & Computers*, is particularly interesting because it reflects the interests of our early donors who are mostly Computer Science graduate students at Indiana University. It contains pages like the graduate students announcement board, the departmental site, and profiles from `thefacebook.com`.

To visually analyze the structure of the network we use *LaNet-vi*,[2] a layout algorithm based on k-core decomposition [2]. Like other visualization tools, *LaNet-vi* assumes an unweighted network and thus we must first use a threshold on the weights (similarities) to select edges. However, as shown in Figure 3, the similarity is distributed broadly, over three orders of magnitudes. This suggests that any threshold value on the similarity weights will lead to a loss of important structural information. Figure 3 also suggests that $s \approx 0.03$ is the critical region in which the weight distribution transitions into a power-law tail, and therefore this is the critical region where we expect to find interesting threshold values for the similarity. Therefore we visualize in Figure 4 three versions of the

[2] `xavier.informatics.indiana.edu/lanet-vi/`

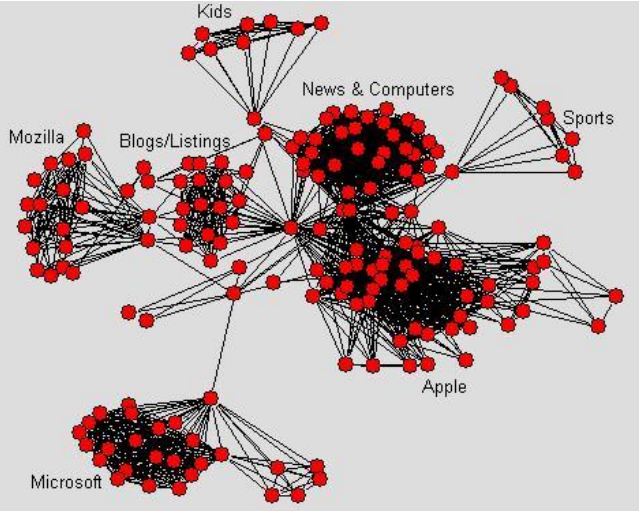


Figure 2: Graph topology generated using Pajek. Nodes displayed are those with at least two edges and edges with $s < 0.04$ have been removed. Labels are added by hand to reflect cluster content.

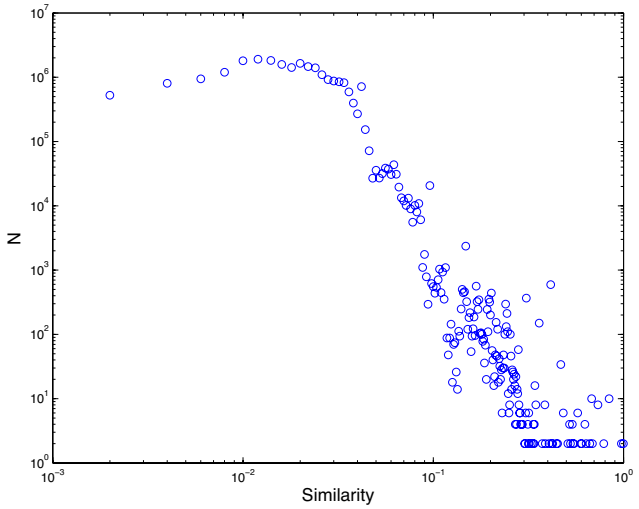


Figure 3: Distribution of the GiveALink link weights, i.e., the similarities $s > 0$ among all the pairs or URLs.

network corresponding to thresholds in this critical region. For high threshold values ($s > 0.04$) we can identify different clusters corresponding to those in Figure 2. As we consider more edges, this cluster structure is gradually lost as we gain more information on the topology. For $s > 0.02$ the network becomes very hierarchical and layered; the number of cores reaches 90. We also note that the degree of nodes appears to be strongly correlated with their centrality; intuitively general pages are similar to more pages than specific ones. Due to the sparsity of the similarity matrix and the distribution of s (cf. Figure 3), the majority of nodes appear isolated at these threshold values and are not displayed; we see at most 419 of the 22,065 URLs ($s > 0.02$).

Another way to analyze the GiveALink similarity network is to compare it with other ways to estimate semantic similarity between the bookmarked Web pages. In prior work [11, 9] we have compared similarity measures obtained from content (text) and hyper-

Table 2: Spearman correlation coefficients between different similarity measures. All differences are statistically significant.

	Link	GiveALink	Semantic
Content	0.055	0.045	0.138
Link		0.026	0.040
GiveALink			0.082

link similarity, and a *semantic* similarity measure obtained from the ODP. As mentioned in § 2, the correlations between these measures across all pairs of pages in the ODP are quite low. To compare the GiveALink similarity, we focus on the intersection between GiveALink and the ODP, i.e., we retain the 1,496 bookmarked pages that are also in the ODP. This yields the correlations shown in Table 2. Since the semantic similarity is based on the golden standard of manual classification and validated by user assessments [9], we can use it as a reference. From this perspective, looking at the third column in Table 2, we find that GiveAlink is a worse predictor than text similarity, but a better predictor than link similarity. However all correlations are very low, suggesting that the collaborative filtering relationship in the GiveALink similarity yields yet a different kind of information than the other measures.

Figure 5 reinforces this view, showing that the topologies of the four similarity networks are qualitatively different. The content-based network is more regular, with correlated degree and centrality. The link-based and semantic networks are less regular and correlated. The GiveALink network has intermediate regularity and correlation between degree and centrality, but is very layered with 92 cores. Again we conclude that the different similarities provide us with different information about relationships among pages.

5. APPLICATIONS

5.1 Search

The pivotal application of the GiveALink project is a search system that allows users to explore the bookmark collection. Figure 6 shows its interface and the results from a simple query. When the user provides a query URL, the system looks for other URLs that have high bookmark similarity to the query, according to our similarity matrix s. Search results can be ranked according to a combination of four different measures: bookmark similarity and three additional ranking measures described below. If the user picks several ranking measures, then results are ranked by the product of their values. If the user does not pick any ranking measure, results are ranked by bookmark similarity to the query.

The GiveALink database is currently quite small and it is often the case that it will not contain the query URL provided by the user. If we do not have the query URL, it is impossible to estimate similarity between it and other URLs in the collection based on our similarity measures. In this case we resort to help from a search engine: we submit the query URL to, say, the Google API and search for similar sites. From the top ten results that Google returns, we pick those that are in our database and expand the resulting set with additional sites from our database similar to them. Finally, we rank the results in the same way as before, using a combination of similarity and ranking measures. Note that we only return URLs that are in our database, and therefore the similarity and ranking values are known for all of them. The additional URLs from our database carry similarity and ranking values with respect to the Google result that generated them.

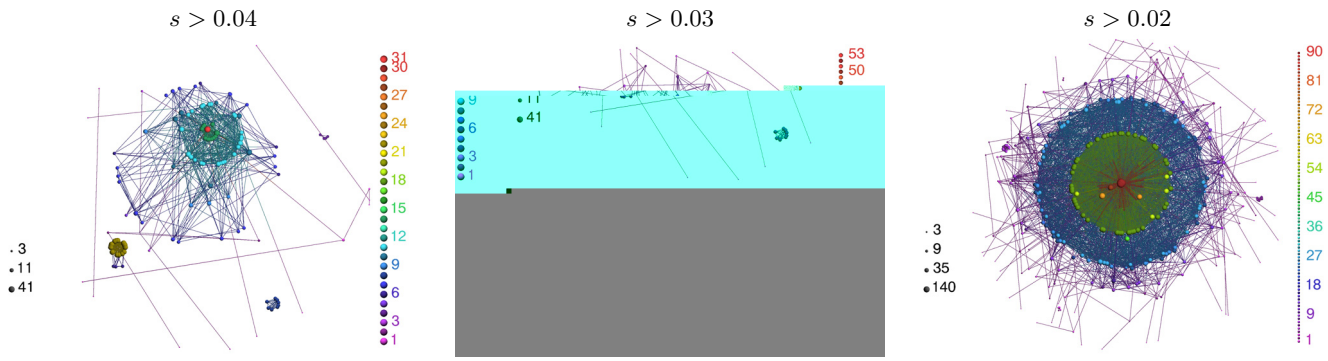


Figure 4: Visualizations of the GiveALink similarity network with different thresholds on edge weights. The *LaNet-vi* tool uses the size of a node to represent its degree (left legend) and colors to represent cores (right legend). Higher-numbered cores are the more internal components of the networks.

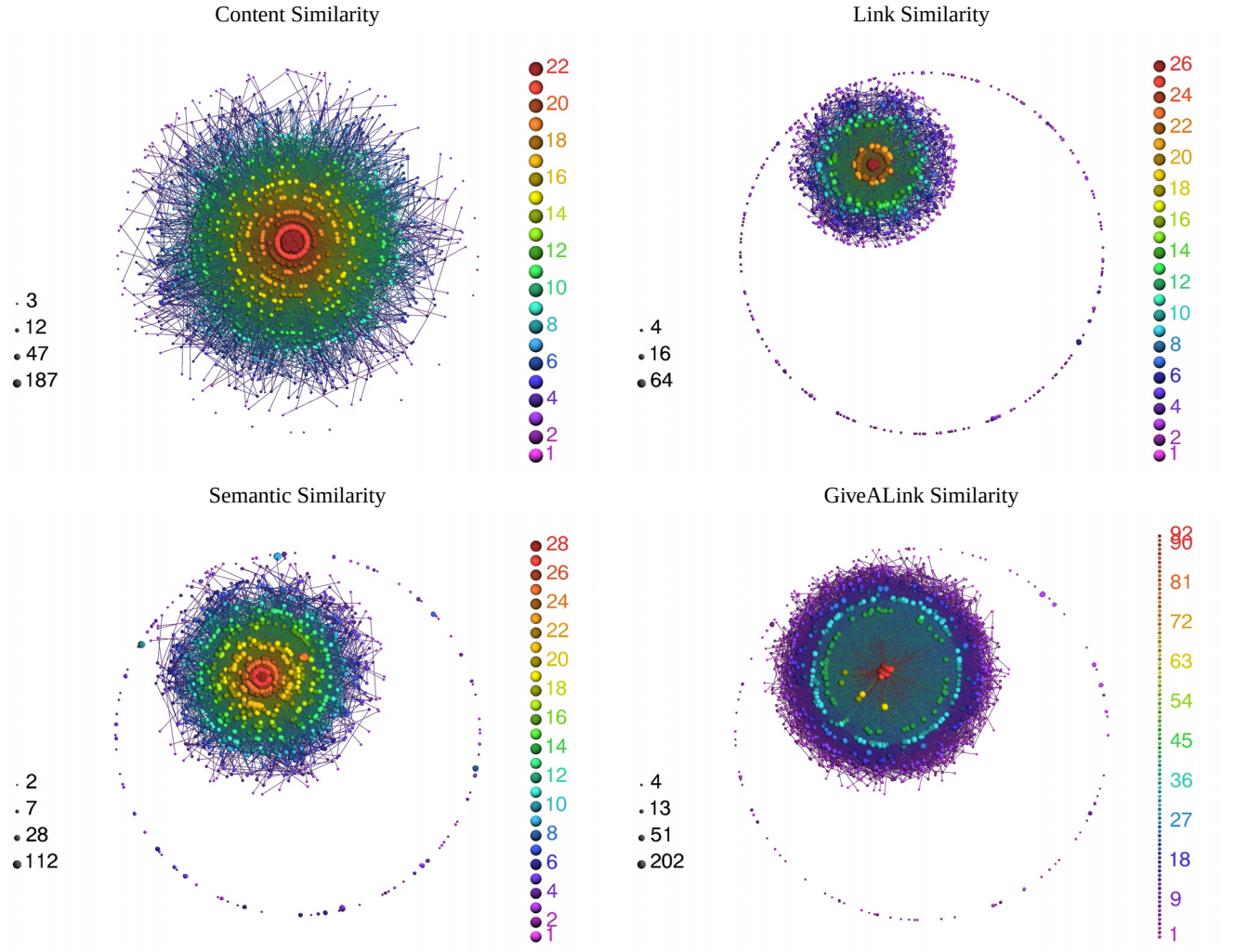


Figure 5: Visualizations of four similarity networks using the 1,496 pages that appear in both GiveALink and the ODP. For each similarity, a threshold is chosen to maximize the size of the network that can be visualized with the public *LaNet-vi* tool.

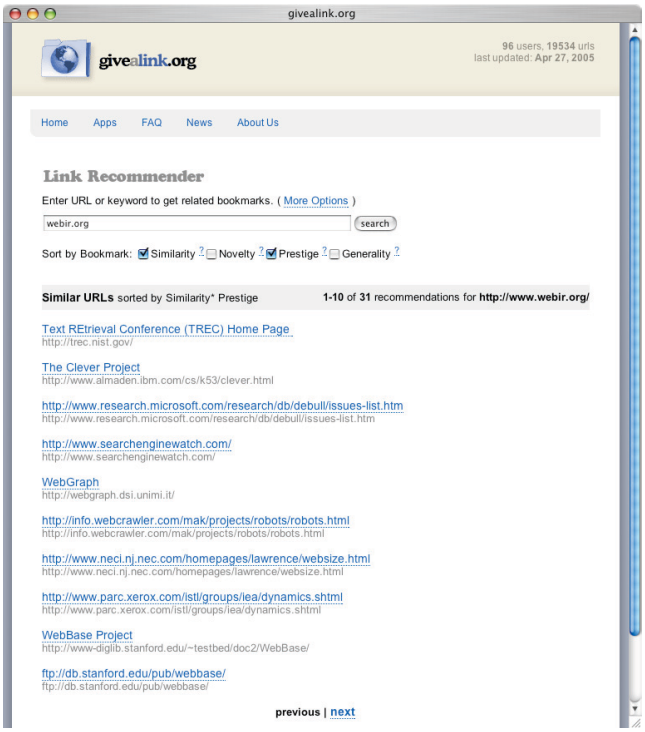


Figure 6: Screen shot of the GiveALink search system.

Instead of providing a query URL, users also have the option of typing in keywords. The interface of this system mimics the familiar interface of search engines. The query is submitted to a search engine API and the top ten results are processed in the way described above. As our bookmark collection grows, our goal is to make this system independent of external search engines. We plan to match the query keywords against the descriptions and titles that users enter for the URLs in their bookmark files. It would also be possible to crawl and index the donated URLs, although at present this is not a research direction we are pursuing.

The similarity matrix described above provides one way of ranking search results: according to bookmark similarity s to the query URL. We also derive three other ranking measures that we refer to as *generality, prestige,* and *novelty.* They provide more ways to rank query results by taking into account aspects of the global associative semantics of the bookmark network. Generality and prestige provide total orders for the URLs in our collection that are independent of the user query. Novelty combines bookmark similarity of search results to the user query with an aspect of the global network, namely semi-metric distances.

5.1.1 Prestige

Prestige is a recursive measure inspired by Google's PageRank[3] — the prestige of a URL is tied to the prestige of its neighbors in the similarity graph. The difference between our prestige and PageRank is that PageRank is computed on a directed, unweighted graph where edges represent hyperlinks. Prestige is computed on an undirected, weighted graph in which the weights of edges represent social similarity s as defined in our similarity matrix. The iterative process is defined as follows: at time step $t = 1$, we give all of the URLs prestige values equal to 1. For each consecutive step, the prestige of node i at time $t + 1$ is

$$P_i(t+1) = (1-\alpha) + \alpha \times \sum_j \frac{s(i,j) \times P_j(t)}{\sum_k s(j,k)}. \qquad (3)$$

The computation continues until the prestige values converge. We use $\alpha = 0.85$.

5.1.2 Generality

Generality is the term selected to describe to our non-technical users the *centrality* of a URL node in the similarity matrix. The centrality of a URL is the average of the shortest-path similarities s_{max} between this node and every other node. A URL with high centrality is one that is very similar to all other URLs in our collection.

Calculating centrality requires the computation of the similarity between all pairs of nodes according to all of the paths between them. There are many ways in which this can be done. One possible approach is to compute the similarity on a given path as the product of the similarity values along all edges in the path. For example, if URLs x and y are connected by a path $x \rightsquigarrow y$ that goes through z, where $s(x,z) = 0.5$ and $s(z,y) = 0.4$, then the similarity between x and y on that path is $s(x \rightsquigarrow y) = 0.5 \times 0.4 = 0.2$. Although this approach is rather intuitive, it is too aggressive for computing similarities [14].

In our system, we convert similarity values to distances, then we compute shortest-path distances using Floyd-Warshall's algorithm [4], and finally we convert these values back into shortest-path similarity values. To convert between similarity and distance values, we use the following formula:

$$dist(x,y) = \frac{1}{s(x,y)} - 1. \qquad (4)$$

Note that when similarity is equal to 1, distance is equal to 0, and when similarity is equal to 0, distance is infinity. Thus the closer two URLs are, the higher their similarity is. The distance along a given path is the sum of the distances along all edges in the path. The shortest-path similarity between two pages is thus defined as

$$s_{max}(x,y) = \left[1 + \min_{x \rightsquigarrow y} \sum_{(u,v) \in x \rightsquigarrow y} \left(\frac{1}{s(u,v)} - 1\right)\right]^{-1}. \qquad (5)$$

5.1.3 Novelty

A natural observation to make, once we have computed the all-pairs shortest-path similarities, is that for some pairs of URLs the indirect shortest-path similarity s_{max} is higher than the direct edge similarity s. There are pairs of URLs, x and y, where $s(x,y)$ is relatively low, but if both x and y are very similar to a third URL z, then their shortest-path similarity $s_{max}(x,y)$ could be much higher. This phenomenon, known as semi-metric behavior [13], is very valuable for a recommendation system because it reveals similarity that is implied by the global associative semantics but has not yet been discovered by individual bookmark users. If used in addition to a direct similarity measure, it empowers the recommendation system to not only generate recommendations that are natural and obvious, but also ones that are unexpected and could inspire users to broaden and deepen their interests.

We attempt to exploit semi-metric behaviors by a novelty measure defined as

$$novelty(x,y) = \begin{cases} \frac{s_{max}(x,y)}{s(x,y)} & \text{if } s(x,y) > 0 \\ \frac{s_{max}(x,y)}{s_{min}} & \text{if } s(x,y) = 0 \end{cases} \qquad (6)$$

where s_{min} is the smallest non-zero similarity value, $s_{min} = \min_{s(x',y')>0} s(x', y')$. This measure is similar to one of the semi-metric ratios introduced by Rocha [13]. For purposes of recommendation we are only interested in pairs of URLs where $novelty(x, y) > 1$, i.e. the indirect similarity is higher than the direct similarity. As the gap grows, the novelty value grows as well.

We call this measure novelty because, when a user submits query x and our search engine returns answer y, where $s(x, y)$ is low but $s_{max}(x, y)$ is high, then y is a valid recommendation and is novel with respect to the measured associations of any one user. Indeed, the indirect associations captured by the novelty ratio are a global property of the network and cannot be locally measured from direct association levels [13]. If the user chooses to rank search results by novelty (or some combination of measures that include novelty), then the recommendations that are non-trivial and unexpected will be ranked higher.

5.2 Recommendation

A natural extension of ranking search results by novelty is a recommendation system that is roughly equivalent to searching by novelty. In the standard search system, results are generated by mining the database for URLs that have high bookmark similarity to the user query. Thus the standard search system is essentially "search by similarity." On the other hand, in the recommendation system, the results are URLs that have high *novelty* to the user query. Results are generated from different sets of URLs in the two applications. The search system considers all URLs in the database and picks the ones most similar to the query. The recommendation system only considers the URLs that have higher shortest-path similarity than direct similarity to the query, and picks the ones with highest novelty.

The two types of systems, search and recommendation, address different information needs. The search system provides additional information that is highly relevant to the user query; if the user provides a URL as the query term, the search results will perhaps expand on the knowledge already contained in the query URL. The recommendation system, on the other hand, provides different information that relates to the query in a non-trivial way. Rather than presenting similar information, recommendation results will provide pages that address the same questions from a different perspective: a different domain of knowledge, perhaps a different time period or geographical location. Thus the recommendation system could inspire collaboration between groups of users who do not yet realize they have similar interests.

5.3 Case Study

In Table 3 we illustrate the different ranking measures using the results of searching for `apple.com`. For comparison, we also present the pages that Google deems "similar," by submitting a `related:apple.com` query to Google. Google does not disclose how similar pages are identified[3] but it is safe to assume that a combination of text and link analysis is employed. As with the clustering seen in 2, these results are highly influenced by the interests of the system's early users as exemplified by the appearance of the IU computer science graduate student Web-board in the top ten results of prestige.

The results are quite exciting with regard to the obvious differences between rankings. Whereas nine of the ten results Google provides are corporate homepages, our Similarity ranking clearly values sites of practical interest to Mac users. Novelty appears to work as intended, revealing potentially relevant sites not listed among the other rankings.

[3] `www.google.com/help/features.html#related`

6. CONCLUSIONS

In summary, we presented GiveALink, a public site where users donate their bookmarks to the Web community. The proposed similarity measure for URLs takes advantage of both the hierarchical structure of bookmark files and collaborative filtering across users. The social bookmark network induced by the similarity measure seems to display meaningful clusters but is qualitatively different, and weakly correlated with, other Web page networks built from similarity measures extracted from content, links, and classification ontologies. We also introduced a search and recommendation system with prestige, generality, and novelty ranking measures extracted from the similarity data.

An obvious advantage of our system when compared to traditional search engines is that we can calculate similarity and make guesses about the topic of a page without having to crawl it. Traditional search engines use text analysis tools (like cosine similarity) to estimate the relevance of a URL with respect to the user query. Our similarity measure, however, does not depend on the content of the page at all and we can recommend movie feeds, javascripts, URLs containing images only, files in various formats, and so on without having the means to access their contents.

Regarding coverage, we note that not all the URLs in our collection are known to Google. For example Google crawled relatively few of the HTTPS pages that people donated as bookmarks; this might be due to Robot Exclusion policies. In addition, we suspect that some users bookmark pages that are not linked from other pages on the Web and thus are invisible to search engine crawlers.

Coverage is also the main current challenge for GiveALink. The sparsity of the bookmark similarity network is due to a small number of donors. We will attempt to achieve critical mass by removing barriers and creating incentives for users to donate bookmarks. For example, users must be convinced that their privacy will be honored by the system, and their use of GiveALink can be facilitated by Web services for search and recommendation.

Having looked at item-to-item (bookmark-to-bookmark) collaborative filtering using bookmark files, an obvious next step is person-to-person collaborative filtering. To this end there are two applications on which we plan to focus our future efforts:

Profile Based Recommendations This system will recommend sites based on the bookmarks that a particular user has submitted. Such recommendations are offered without a query being specified by the user. The process is such that users are clustered, and the most common or interesting URLs not found in a particular user's collection, but found among related users collections, are offered as recommendations.

Personalized Search Personalized search, like user profile recommendations, will be based on bookmarks submitted by an individual and the clustering of like individuals. Unlike user profile recommendations, the results are query-dependent. The results are a subset of "regular" search results that are relevant to the specific cluster of users to which the current user belongs. Users might also have multiple profiles based on subsets of their bookmark collection.

We make all of our non-personal data freely available to the Web research community and general Internet users in the hope that it will foster the development of many novel Web mining techniques. Our similarity matrix, as well as generality and prestige scores for all bookmarks in our collection, can be downloaded from the project Web site at `www.givealink.org`.

Table 3: Recommendations for `apple.com`.

Similarity	Novelty	Prestige*Similarity	Generality*Similarity	Google (related:apple.com)
Apple–Store	Shockwave.com	CNN	Apple–Support	Apple
Apple–Support	CNET	Apple–Support	Apple–Store	Microsoft
Apple–SW Updates	Warehouse.com/Apple	Apple–Store	MSN	Adobe
Mactopia	Inside Mac Games	Google News	Mactopia	Sun
Office for Mac	MacWindows	MSN	Office for Mac	Macromedia
Apple Livepage	Apple–Product Guide	BBC News	Apple–SW Updates	HP
MSN	MozillaZine	Mapquest	Apple–Products	IBM
Apple–Products	MacToday	Apple–Quicktime	Apple–Hot News	Dell
Apple–Hot News	MacGamer	IU CS Webboard	Apple Livepage	Apple–MacOS X
Apple–MacOS X	MacGaming	Mactopia	Apple–MacOS X	Netscape

Acknowledgments

We would like to thank Luis Rocha for suggesting the novelty ranking measure and the recommendation system; Alessandro Vespignani for discussing the evaluation of the networks; Bill English for designing the GiveALink Web site; Rob Henderson and Jacob Ratkiewicz for invaluable technical help; and the NaN group at Indiana University for helpful discussions throughout the development of the project. This work was funded in part by NSF Career Grant IIS-0348940 to FM.

7. REFERENCES

[1] David Abrams, Ronald Baecker, and Mark H. Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *CHI*, pages 41–48, 1998.

[2] Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. k-core decomposition: A tool for the visualization of large scale networks. Computing Research Repository (CoRR), http://arxiv.org/abs/cs.NI/0504107, 2005.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorightms, 2nd ed.* MIT Press, 2001.

[5] Prasanna Ganesan, Hector Garcia-Molina, and Jennifer Widom. Exploiting hierarchical domain structure to compute similarity. In *ACM Trans. Inf. Syst. 21(1)*, pages 64–93, 2003.

[6] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (I): A general review. *D-Lib Magazine*, 11(4):doi:10.1045/april2005–hammond, 2005.

[7] Markus Jakobsson and Filippo Menczer. Web forms and untraceable ddos attacks. In S. Huang, D. MacCallum, and D.Z. Du, editors, *Network Security*, Forthcoming in June 2005.

[8] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[9] Ana Maguitman, Filippo Menczer, Heather Roinestad, and Alessandro Vespignani. Algorithmic detection of semantic similarity. In *Proc. WWW2005*, 2005.

[10] Bruce McKenzie and Andy Cockburn. An empirical analysis of web page revisitation. In *Proc. of the 34th Hawaii International Conference on System Sciences*, 2001.

[11] Filippo Menczer. Mapping the semantics of web text and links. *IEEE Internet Computing*, 9(3):27–36, May/June 2005.

[12] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstorm, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[13] Luis M. Rocha. Semi-metric behavior in document networks and its application to recommendation systems. In V. Loia, editor, *Soft Computing Agents: A New Perspective for Dynamic Information Systems*, pages 137–163. International Series Frontiers in Artificial Intelligence and Applications. IOS Press, 2002.

[14] Luis M. Rocha. Personal communication, 2005.

[15] Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommender algorithms for e-commerce. In *Proceedings of the 2nd ACM E-Commerce Conference (EC'00)*, 2000.

[16] Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[17] Brent Smith, Greg Linden, and Jeremy York. Amazon.com recommendations: item-to-item collaborative filtering. In *Internet Computing, IEEE*, 2003.

[18] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. *Advances in Cryptology: Eurocrypt*, 2003.

Group and Topic Discovery from Relations and Text

Xuerui Wang, Natasha Mohanty, Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{xuerui,nmohanty,mccallum}@cs.umass.edu

ABSTRACT

We present a probabilistic generative model of entity relationships and textual attributes that simultaneously discovers groups among the entities and topics among the corresponding text. Block-models of relationship data have been studied in social network analysis for some time. Here we simultaneously cluster in several modalities at once, incorporating the words associated with certain relationships. Significantly, joint inference allows the discovery of groups to be guided by the emerging topics, and vice-versa. We present experimental results on two large data sets: sixteen years of bills put before the U.S. Senate, comprising their corresponding text and voting records, and 43 years of similar data from the United Nations. We show that in comparison with traditional, separate latent-variable models for words or Blockstructures for votes, the Group-Topic model's joint inference improves both the groups and topics discovered.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms

Algorithms, experimentation

Keywords

Graphical models, text modeling, relational learning

1. INTRODUCTION

Research in the field of social network analysis (SNA) has led to the development of mathematical models that discover patterns in interaction between entities [21, 5, 14]. One of the objectives of SNA is to detect salient groups of entities. Group discovery has many applications, such as understanding the social structure of organizations [6] or native tribes [8], uncovering criminal organizations [19], and modeling large-scale social networks in Internet services such as Friendster.com or LinkedIn.com.

Social scientists have conducted extensive research on group detection, especially in fields such as anthropology [8] and political science [11, 7]. Recently, statisticians and computer scientists have begun to develop models that specifically discover group memberships [15, 3, 17, 13]. One such model is the stochastic Blockstructures model [17], which discovers the latent structure, groups or classes based on pair-wise relation data. A particular relation holds between a pair of entities (people, countries, organizations, etc.) with some probability that depends only on the class (group) assignments of the entities. The relations between all the entities can be represented with a directed or undirected graph. The class assignments can be inferred from a graph of observed relations or link data using Gibbs sampling [17]. This model is extended in [13] to automatically select an arbitrary number of groups by using a Chinese Restaurant Process prior.

The aforementioned models discover latent groups only by examining whether one or more relations exist between a pair of entities. The Group-Topic (GT) model presented in this paper, on the other hand, considers not only the relations between objects but also the attributes of the relations (for example, the text associated with the relations) when assigning group membership.

The GT model can be viewed as an extension of the stochastic Blockstructures model [17, 13] with the key addition that group membership is conditioned on a latent variable associated with the attributes of the relation. In our experiments, the attributes of relations are words, and the latent variable represents the topic responsible for generating those words. Unlike previous methods, our model captures the *(language) attributes* associated with interactions between entities, and uses distinctions based on these attributes to better assign group memberships.

Consider a legislative body and imagine its members forging alliances (forming groups), and voting accordingly. However, different alliances arise depending on the topic of the resolution up for a vote. For example, one grouping of the legislators may arise on the issue of taxation, while a quite different grouping may occur for votes on foreign trade. Similar patterns of topic-based affiliations would arise in other types of entities as well, e.g., research paper co-authorship relations between people and citation relations between papers, with words as attributes on these relations.

In the GT model, the discovery of groups is guided by the emerging topics, and the discovery of topics is guided

LinkKDD 2005 Chicago, IL, USA

by emerging groups. Both modalities are driven by the common goal of increasing data likelihood. Consider the voting example again; resolutions that would have been assigned the same topic in a model using words alone may be assigned to different topics if they exhibit distinct voting patterns. Distinct word-based topics may be merged if the entities vote very similarly on them. Likewise, multiple different divisions of entities into groups are made possible by conditioning them on the topics.

The importance of modeling the *language* associated with interactions between people has recently been demonstrated in the Author-Recipient-Topic (ART) model [16]. In ART the words in a message between people in a network are generated conditioned on the author, recipients and a set of topics that describes the message. The model thus captures both the network structure within which the people interact as well as the language associated with the interactions. In experiments with Enron and academic email, the ART model is able to discover role similarity of people better than SNA models that consider network connectivity alone. However, the ART model does not explicitly capture groups formed by entities in the network.

The GT model simultaneously clusters entities to groups and clusters words into topics, unlike models that generate topics solely based on word distributions such as Latent Dirichlet Allocation [4]. In this way the GT model discovers salient topics relevant to relationships between entities in the social network—topics which the models that only examine words are unable to detect.

We demonstrate the capabilities of the GT model by applying it to two large sets of voting data: one from US Senate and the other from the General Assembly of the UN. The model clusters voting entities into coalitions and simultaneously discovers topics for word attributes describing the relations (bills or resolutions) between entities. We find that the groups obtained from the GT model are significantly more cohesive (p-value $< .01$) than those obtained from the Blockstructures model. The GT model also discovers new and more salient topics in both the Senate and UN datasets—in comparison with topics discovered by only examining the words of the resolutions, the GT topics are either split or joined together as influenced by the voters' patterns of behavior.

2. GROUP-TOPIC MODEL

The Group-Topic Model is a directed graphical model that clusters entities with relations between them, as well as attributes of those relations. The relations may be either directed or undirected and have multiple attributes. In this paper, we focus on undirected relations and have words as the attributes on relations.

In the generative process for each event (an interaction between entities), the model first picks the topic t of the event and then generates all the words describing the event where each word is generated independently according to a multinomial (discrete) distribution ϕ_t, specific to the topic t. To generate the relational structure of the network, first the group assignment, g_{st} for each entity s is chosen conditionally from a particular multinomial (discrete) distribution θ_t over groups for each topic t. Given the group assignments on an event b, the matrix $V^{(b)}$ is generated where each cell $V_{ij}^{(b)}$ represents if the groups of two entities (i and j) behaved the same or not during the event b, (e.g., voted the same or not on a bill). Each element of V is sampled from a binomial (Bernoulli) distribution $\gamma_{g_i g_j}^{(b)}$. Our notation is summarized in Table 1, and the graphical model representation of the model is shown in Figure 1.

SYMBOL	DESCRIPTION
g_{it}	entity i's group assignment in topic t
t_b	topic of an event b
$w_k^{(b)}$	the kth token in the event b
$V_{ij}^{(b)}$	entity i and j's groups behaved same (1) or differently (2) on the event b
S	number of entities
T	number of topics
G	number of groups
B	number of events
V	number of unique words
N_b	number of word tokens in the event b
S_b	number of entities who participated in the event b

Table 1: Notation used in this paper

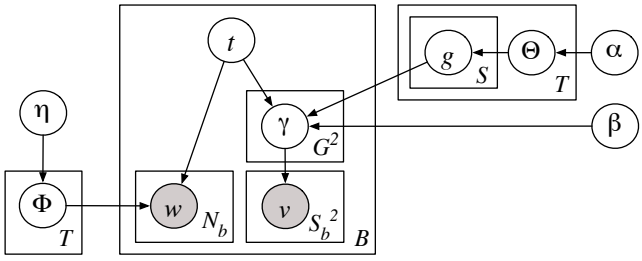


Figure 1: The Group-Topic model

Without considering the topic of an event, or by treating all events in a corpus as reflecting a single topic, the simplified model (only the right part of Figure 1) becomes equivalent to the stochastic Blockstructures model [17]. To match the Blockstructures model, each event defines a relationship, *e.g.*, whether in the event two entities' groups behave the same or not. On the other hand, in our model a relation may have multiple attributes (which in our experiments are the words describing the event, generated by a per-topic multinomial (discrete) distribution).

When we consider the complete model, the dataset is dynamically divided into T sub-blocks each of which corresponds to a topic. The complete GT model is as follows,

$$
\begin{aligned}
t_b &\sim \text{Uniform}(\frac{1}{T}) \\
w_{it}|\phi_t &\sim \text{Multinomial}(\phi_t) \\
\phi_t|\eta &\sim \text{Dirichlet}(\eta) \\
g_{it}|\theta_t &\sim \text{Multinomial}(\theta_t) \\
\theta_t|\alpha &\sim \text{Dirichlet}(\alpha) \\
V_{ij}^{(b)}|\gamma_{g_i g_j}^{(b)} &\sim \text{Binomial}(\gamma_{g_i g_j}^{(b)}) \\
\gamma_{gh}^{(b)}|\beta &\sim \text{Beta}(\beta).
\end{aligned}
$$

We want to perform joint inference on (text) attributes and relations to obtain topic-wise group memberships. Since inference can not be done exactly on such complicated probabilistic graphical models, we employ Gibbs sampling to conduct inference. Note that we adopt conjugate priors in our

setting, and thus we can easily integrate out θ, ϕ and γ to decrease the uncertainty associated with them. This simplifies the sampling since we do not need to sample θ, ϕ and γ at all, unlike in [17]. In our case we need to compute the conditional distribution $P(g_{st}|\mathbf{w}, \mathbf{V}, \mathbf{g}_{-st}, \mathbf{t}, \alpha, \beta, \eta)$ and $P(t_b|\mathbf{w}, \mathbf{V}, \mathbf{g}, \mathbf{t}_{-b}, \alpha, \beta, \eta)$, where $\mathbf{g}_{-st}$ denotes the group assignments for all entities except entity s in topic t, and $\mathbf{t}_{-b}$ represents the topic assignments for all events except event b. Beginning with the joint probability of a dataset, and using the chain rule, we can obtain the conditional probabilities conveniently. The derivations are provided in detail in Appendix A. In our setting, the relationship we are investigating is always symmetric, so we do not distinguish R_{ij} and R_{ji} in our derivations (only $R_{ij}(i \leq j)$ remain). Thus

$$\begin{aligned} & P(g_{st}|\mathbf{V}, \mathbf{g}_{-st}, \mathbf{w}, \mathbf{t}, \alpha, \beta, \eta) \\ \propto \; & \frac{\alpha_{g_{st}} + n_{tg_{st}} - 1}{\sum_{g=1}^{G}(\alpha_g + n_{tg}) - 1} \prod_{b=1}^{B} \Bigg(I(t_b = t) \\ & \times \prod_{h=1}^{G} \frac{\prod_{k=1}^{2} \prod_{x=1}^{d_{g_{st}hk}^{(b)}} \left(\beta_k + m_{g_{st}hk}^{(b)} - x\right)}{\prod_{x=1}^{\sum_{k=1}^{2} d_{g_{st}hk}^{(b)}} \left((\sum_{k=1}^{2}(\beta_k + m_{g_{st}hk}^{(b)}) - x\right)} \Bigg), \end{aligned}$$

where n_{tg} represents how many entities are assigned into group g in topic t, c_{tv} represents how many tokens of word v are assigned to topic t, $m_{ghk}^{(b)}$ represents how many times group g and h vote same ($k = 1$) and differently ($k = 2$) on event b, $I(t_b = t)$ is an indicator function, and $d_{g_{st}hk}^{(b)}$ is the increase in $m_{g_{st}hk}^{(b)}$ if entity s were assigned to group g_{st} than without considering s at all (if $I(t_b = t) = 0$, we ignore the increase in event b). Furthermore,

$$\begin{aligned} & P(t_b|\mathbf{V}, \mathbf{g}, \mathbf{w}, \mathbf{t}_{-b}, \alpha, \beta, \eta) \\ \propto \; & \frac{\prod_{v=1}^{V} \prod_{x=1}^{e_v^{(b)}} (\eta_v + c_{t_b v} - x)}{\prod_{x=1}^{\sum_{v=1}^{V} e_v^{(b)}} \left(\sum_{v=1}^{V}(\eta_v + c_{t_b v}) - x\right)} \\ & \times \prod_{g=1}^{G} \prod_{h=g}^{G} \frac{\prod_{k=1}^{2} \Gamma(\beta_k + m_{ghk}^{(b)})}{\Gamma(\sum_{k=1}^{2}(\beta_k + m_{ghk}^{(b)}))}, \end{aligned}$$

where $e_v^{(b)}$ is the number of tokens of word v in event b. Note that $m_{ghk}^{(b)}$ is not a constant and changes with the assignment of t_b since it influences the group assignments of all entities that vote on event b.

The GT model uses information from two different modalities. In general, the likelihood of the two modalities is not directly comparable, since the number of occurrences of each type may vary greatly (e.g., there may be far more pairs of voting entities than word occurrences). Thus we use a weighting parameter to rescale the likelihoods from different modalities, as is also common in speech recognition when the acoustic and language models are combined.

3. RELATED WORK

There has been a surge of interest in models that describe relational data, or relations between entities viewed as links in a network, including recent work in group discovery. One such algorithm, presented by Bhattacharya and Getoor [3], is a bottom-up agglomerative clustering algorithm that partitions links in a network into clusters by considering the change in likelihood that would occur if two clusters were merged. Once the links have been grouped, the entities connected by the links are assigned to groups.

Another model due to Kubica et al. [15] considers both link evidence and attributes on entities to discover groups. The Group Detection Algorithm (GDA) uses a Bayesian network to group entities from two datasets, demographic data describing the entities and link data. Unlike our model, neither of these models [3, 15] consider attributes associated with the links between the entities. The model presented in [15] considers attributes of an entity rather than attributes of relations between entities.

The central theme of GT is that it simultaneously clusters entities and attributes on relations (words). There has been prior work in clustering different entities simultaneously, such as information theoretic co-clustering [9], and multi-way distributional clustering using pair-wise interactions [2]. However, these models do not also cluster attributes based on interactions between entities in a network.

In our model, group membership defines pair-wise relations between nodes. The GT model is an enhancement of the stochastic Blockstructures model [17] and the extended model of Kemp et al. [13] as it takes advantage of information from different modalities by conditioning group membership on topics. In this sense, the GT model draws inspiration from the Role-Author-Recipient-Topic (RART) model [16]. As an extension of ART model, RART clusters together entities with similar roles. In contrast, the GT model presented here clusters entities into groups based on their relations to other entities.

Exploring the notion that the behavior of an entity can be explained by its (hidden) group membership, Jakulin and Buntine [12] develop a discrete PCA model for discovering groups. In the model each entity can belong to each of the k groups with a certain probability, and each group has its own specific pattern of behaviors. Therefore, an entity's behavior depends on the probability of belonging to a group and the probability that the group has that behavior. They apply this model to voting data in the 108th US Senate where the behavior of an entity is its vote on a resolution. A similar model is developed in [18] that examines group cohesion and voting similarity in the Finnish Parliament. We apply our GT model also to voting data. However, unlike [12, 18], since our goal is to cluster entities based on the similarity of their voting patterns, we are only interested in whether a pair of entities voted the same or differently, not their actual yes/no votes. Two resolutions on the same topic may differ only in their goal (e.g., increasing vs. decreasing budget), thus the actual votes on one could be the converse of votes on the other. However, pairs of entities who vote the same on one resolution would tend to vote same on the other resolution. To capture this, we model relations as *agreement* between entities, not the yes/no vote itself. This kind of "content-ignorant" feature is similarly found in some work on web log clustering [1].

There has been a considerable amount of previous work in understanding voting patterns [10, 11, 7], including research on voting cohesion of countries in the EU parliament [11] and partisanship in roll call voting [7]. In these models roll call data are used to estimate *ideal points* of a legislator (which refers to a legislator's preferred policy in the Euclidean space of possible policies). The models assume that each vote in the roll call data is independent of the remaining votes, i.e., each individual is not connected to anyone else who is voting.

Datasets	Avg. AI for GT	Avg. AI for Baseline	p-value
Senate	0.8294	0.8198	$< .01$
UN	0.8664	0.8548	$< .01$

Table 2: Average AI for GT and Baseline for both Senate and UN datasets. The group cohesion in GT is significantly better than in baseline.

However, in reality, legislation is shaped by the coalitions formed by like-minded legislators. The GT model attempts to capture this interaction.

4. EXPERIMENTAL RESULTS

We present experiments applying the GT model to the voting records of members of two legislative bodies: the US Senate and the UN General Assembly.

For comparison, we present the results of a baseline method that first uses a mixture of unigrams to discover topics and associate a topic with each resolution, and then runs the Blockstructures model [17] separately on the resolutions assigned to each topic. This baseline approach is similar to the GT model in that it discovers both groups and topics, and has different group assignments on different topics. However, whereas the GT model performs joint inference simultaneously, the baseline performs inference serially. Note that our baseline is still more powerful than the Blockstructures models, since it models the topic associated with each event, and allows the creation of distinct groupings dependent on different topics.

In this paper, we are interested in the quality of both the groups and the topics. In the political science literature, group cohesion is quantified by the *Agreement Index (AI)* [12, 18], which measures the similarity of votes cast by members of a group during a particular roll call. The AI for a particular group on a given roll call i is based on the number of group members that vote Yes(y_i), No(n_i) or Abstain(a_i) in the roll call i. Higher AI index means better cohesion.

$$AI_i = \frac{\max\{y_i, n_i, a_i\} - \frac{y_i+n_i+a_i-\max\{y_i,n_i,a_i\}}{2}}{y_i + n_i + a_i}$$

The Blockstructures model assumes that members of a legislative body have the same group affiliations irrespective of the topic of the resolution on vote. However, it is likely that members form their groups based on the topic of the resolution being voted on. We quantify the extent to which a member s switches groups with a *Group Switch Index* (GSI).

$$GSI_s = \sum_{i,j}^{T} \frac{\text{abs}(\vec{s_i} - \vec{s_j})}{|G(s,i)| - 1 + |G(s,j)| - 1}$$

where $\vec{s_i}$ and $\vec{s_j}$ are bit vectors of the length of the size of the legislative body. The k_{th} bit of $\vec{s_i}$ is set if k is in the same group as s on topic i and similarly $\vec{s_j}$ corresponds to topic j. $G(s,i)$ is the group of s on topic i which has a size of $|G(s,i)|$ and $G(s,j)$ is the group of s on topic j. We present entities that frequently change their group alliance according to the topics of resolutions.

The group cohesion using the GT model is found to be significantly greater than the baseline group cohesion under pairwise t-test, as shown in Table 2 for both the Senate and the UN datasets, which indicates that the GT model is better able to capture cohesive groups. We find that nearly every document has a higher Agreement Index across groups using the GT model as compared to the baseline.

Economic	Education	Military Misc.	Energy
federal	education	government	energy
labor	school	military	power
insurance	aid	foreign	water
aid	children	tax	nuclear
tax	drug	congress	gas
business	students	aid	petrol
employee	elementary	law	research
care	prevention	policy	pollution

Table 3: Top words for topics generated with the mixture of unigrams model on the Senate dataset. The headers are our own summary of the topics.

Economic	Education + Domestic	Foreign	Social Security + Medicare
labor	education	foreign	social
insurance	school	trade	security
tax	federal	chemicals	insurance
congress	aid	tariff	medical
income	government	congress	care
minimum	tax	drugs	medicare
wage	energy	communicable	disability
business	research	diseases	assistance

Table 4: Top words for topics generated with the GT model on the Senate dataset. The topics are influenced by both the words and votes on the bills.

4.1 The US Senate Dataset

Our Senate dataset consists of the voting records of Senators in the 101st-109th US Senate (1989-2005) obtained from the Library of Congress THOMAS database. During a roll call for a particular bill, a Senator may respond *Yea* or *Nay* to the question that has been put to vote, else the vote will be recorded as *Not Voting*. We do not consider *Not Voting* as a unique vote since most of the time it is a result of a Senator being absent from the session of the US Senate. The text associated with each resolution is composed of its index terms provided in the database. There are 3423 resolutions in our experiments (we excluded roll calls that were not associated with resolutions). Each bill may come up for vote many times in the U.S. Senate, each time with an attached amendment, and thus many relations may have the same attributes (index terms). Since there are far fewer words than pairs of votes, we adjust the text likelihood to the 5th power (weighting factor 5) in the experiments with this dataset so as to balance its influence during inference.

We cluster the data into 4 topics and 4 groups (cluster sizes are chosen somewhat arbitrarily) and compare the results of GT with the baseline. The most likely words for each topic from the traditional mixture of unigrams model is shown in Table 3, whereas the topics obtained using GT are shown in Table 4. The GT model collapses the topics Education and Energy together into Education and Domestic,

Group 1	Group 3	Group 4
73 Republicans	Cohen(R-ME)	Armstrong(R-CO)
Krueger(D-TX)	Danforth(R-MO)	Garn(R-UT)
Group 2	Durenberger(R-MN)	Humphrey(R-NH)
90 Democrats	Hatfield(R-OR)	McCain(R-AZ)
Chafee(R-RI)	Heinz(R-PA)	McClure(R-ID)
Jeffords(I-VT)	Kassebaum(R-KS)	Roth(R-DE)
	Packwood(R-OR)	Symms(R-ID)
	Specter(R-PA)	Wallop(R-WY)
	Snowe(R-ME)	Brown(R-CO)
	Collins(R-ME)	DeWine(R-OH)
		Thompson(R-TN)
		Fitzgerald(R-IL)
		Voinovich(R-OH)
		Miller(D-GA)
		Coleman(R-MN)

Table 5: Senators in the four groups corresponding to Topic Education + Domestic in Table 4.

Senator	Group Switch Index
Shelby(D-AL)	0.6182
Heflin(D-AL)	0.6049
Voinovich(R-OH)	0.6012
Johnston(D-LA)	0.5878
Armstrong(R-CO)	0.5747

Table 6: Senators that switch groups the most across topics for the 101st-109th Senates

since the voting patterns on those topics are quite similar. The new topic Social Security + Medicare did not have strong enough word coherence to appear in the baseline model, but it has a very distinct voting pattern, and thus is clearly found by the GT model. Thus GT discovers topics that are salient in that they correlate with people's behavior and relations, not simply word co-occurrences.

Examining the group distribution across topics in the GT model, we find that on the topic Economic the Republicans form a single group whereas the Democrats split into 3 groups indicating that Democrats have been somewhat divided on this topic. With regard to Education + Domestic and Social Security + Medicare, Democrats are more unified whereas the Republicans split into 3 groups. The group membership of Senators on Education + Domestic issues is shown in Table 5. We see that the first group of Republicans include a Democratic Senator from Texas, a state that usually votes Republican. Group 2 (majority Democrats) includes Sen. Chafee who is known to be pro-environment and is involved in initiatives to improve education, as well as Sen. Jeffords who left the Republican Party to become an Independent and has championed legislation to strengthen education and environmental protection.

Nearly all the Senators in Group 4 (in Table 5) are advocates for education and many of them have been awarded for their efforts (e.g., Sen. Fitzgerald has been honored by the NACCP for his active role in Early Care and Education, and Sen. McCain has been added to the ASEE list as a *True Hero* in American Education). Sen. Armstrong was a member of the Education committee; Sen. Voinovich and Sen. Symms are strong supporters of early education

Everything Nuclear	Human Rights	Security in Middle East
nuclear	rights	occupied
weapons	human	israel
use	palestine	syria
implementation	situation	security
countries	israel	calls

Table 7: Top words for topics generated from mixture of unigrams model with the UN dataset (1990-2003). Only text information is utilized to form the topics, as opposed to Table 8 where our GT model takes advantage of both text and voting information.

and vocational education, respectively; and Sen. Roth has constantly voted for tax deductions for education. It is also interesting to see that Sen. Miller (D-GA) appears in a Republican group; although he is in favor of educational reforms, he is a conservative Democrat and frequently criticizes his own party—even backing Republican George W. Bush over Democrat John Kerry in the 2004 Presidential election.

Many of the Senators in Group 3 have also focused on education and other domestic issues such as energy, however, they often have a more liberal stance than those in Group 4, and come from states that are historically less conservative. Senators Hatfield, Heinz, Snowe, Collins, Cohen and others have constantly promoted pro-environment energy options with a focus on renewable energy, while Sen. Danforth has presented bills for a more fair distribution of energy resources. Sen. Kassebaum is known to be uncomfortable with many Republican views on domestic issues such as education, and has voted against voluntary prayer in school. Thus, both Groups 3 and 4 differ from the Republican core (Group 2) on domestic issues, and also differ from each other.

The Senators that switch groups the most across topics in the GT model are shown in Table 6 based on their GSIs. Sen. Shelby(D-AL) votes with the Republicans on Economic, with the Democrats on Education + Domestic and with a small group of maverick Republicans on Foreign and Social Security + Medicare. Both Sen. Shelby and Sen. Heflin are Democrats from a fairly conservative state (Alabama) and are found to side with the Republicans on many issues.

4.2 The United Nations Dataset

The second dataset involves the voting record of the UN General Assembly [20]. We focus first on the resolutions discussed from 1990-2003, which contain votes of 192 countries on 931 resolutions. If a country is present during the roll call, it may choose to vote *Yes*, *No* or *Abstain*. Unlike the Senate dataset, a country's vote can have one of three possible values instead of two. Because we parameterize agreement and not the votes themselves, this 3-value setting does not require any change to our model. In experiments with this dataset, we use a weighting factor 500 for text (adjusting the likelihood of text by a power of 500 so as to make it comparable with the likelihood of pairs of votes for each resolution). We cluster this dataset into 3 topics and 5 groups (numbers are chosen somewhat arbitrarily).

The most probable words in each topic from the mixture of unigrams model is shown in Table 7. For example, Every-

GROUP ↓	Nuclear Arsenal	Human Rights	Nuclear Arms Race
	nuclear	rights	nuclear
	states	human	arms
	united	palestine	prevention
	weapons	occupied	race
	nations	israel	space
1	Brazil	Brazil	UK
	Columbia	Mexico	France
	Chile	Columbia	Spain
	Peru	Chile	Monaco
	Venezuela	Peru	East-Timor
2	USA	Nicaragua	India
	Japan	Papua	Russia
	Germany	Rwanda	Micronesia
	UK...	Swaziland	
	Russia	Fiji	
3	China	USA	Japan
	India	Japan	Germany
	Mexico	Germany	Italy...
	Iran	UK...	Poland
	Pakistan	Russia	Hungary
4	Kazakhstan	China	China
	Belarus	India	Brazil
	Yugoslavia	Indonesia	Mexico
	Azerbaijan	Thailand	Indonesia
	Cyprus	Philippines	Iran
5	Thailand	Belarus	USA
	Philippines	Turkmenistan	Israel
	Malaysia	Azerbaijan	Palau
	Nigeria	Uruguay	
	Tunisia	Kyrgyzstan	

Table 8: Top words for topics generated from the GT model with the UN dataset (1990-2003) as well as the corresponding groups for each topic (column). The countries listed for each group are ordered by their 2005 GDP (PPP) and only the top 5 countries are shown in groups that have more than 5 members.

thing Nuclear constitutes all resolutions that have anything to do with the use of nuclear technology, including nuclear weapons. Comparing these with topics generated from the GT model shown in Table 8, we see that the GT model splits the discussion about nuclear technology into two separate topics, Nuclear Arsenal which is generally about countries obtaining nuclear weapons and management of nuclear waste, and Nuclear Arms Race which focuses on the arms race between Russia and the US and preventing a nuclear arms race in outer space. These two issues had drastically different voting patterns in the U.N., as can be seen in the contrasting group structure for those topics in Table 8. The countries in Table 8 are ranked by their GDP in 2005.[1] Thus, again the GT model is able to discover salient topics—topics that reflect the voting patterns and coalitions, not simply word co-occurrence alone.

As seen in Table 8, groups formed in Nuclear Arms Race are unlike the groups formed in the remaining topics. These groups map well to the global political situation of that time when, despite the end of the Cold War, there was mutual distrust between Russia and the US with regard to the continued manufacture of nuclear weapons. For missions to outer space and nuclear arms, India was a staunch ally of Russia, while Israel was an ally of the US.

4.2.1 Overlapping Time Intervals

In order to understand changes and trends in topics and groups over time, we run the GT model on resolutions that were discussed during overlapping time windows of 15 years, from 1960-2000, each shifted by a period of 5 years. We consider 3823 unique resolutions in this way. The topics as well as the group distribution for the most dominant topic during each time period are shown in Table 9.

Over the years there is a shift in the topics discussed in the UN, which corresponds well to the events and issues in history. During 1960-1975 the resolutions focused on countries having the right to self-determination, especially countries in Africa which started to gain their freedom during this time. Although this topic continued to be discussed in the subsequent time period, the focus of the resolutions shifted to the role of the UN in controlling nuclear weapons as the Cold War conflict gained momentum in the late 70s. While there were few resolutions condemning the racist regime in South Africa between 1965-1980, this was the topic of many resolutions during 1970-1985—culminating in the UN censure of South Africa for its discriminatory practices.

Other topics discussed during the 70s and early 80s were Israel's occupation of neighboring countries and nuclear issues. The reduction of arms was primarily discussed during 1975-1990, the time period during which the US and Soviet Union had talks about disarmament. During 1980-1995 the central topic of discussion was the Israeli-Palestinian conflict; this time period includes the beginning of the *Intifada* revolt in Palestine and the Gulf War. This topic continued to be important in the next time period (1985-2000), but in the most recent slice (1990-2003, Table 8) it has become a part of a broader topic on human rights by combining other human rights related resolutions that appear as a separate topic during 1985-2000. The human rights issue continues to be the primary topic of discussion during 1990-2003.

Throughout the history of the UN, the US is usually in the same group as Europe and Japan. However, as we can see in Table 9 during 1985-2000, when the Israeli-Palestinian conflict was the most dominant topic, US and Israel form a group of their own separating themselves from Europe. In other topics discussed during 1985-2000, US and Israel are found to be in the same group as Europe and Japan.

Another interesting result of considering the groups formed over the years is that, except for the last time period (1990-2003), countries in eastern Europe such as Poland, Hungary, Bulgaria, etc., form a group along with USSR (Russia). However, in the last time window on most topics they become a part of the group that consists of the western Europe, Japan and the US. This shift corresponds to the end of the communist regimes in these countries that were supported by the Soviet Union. It is also worth mentioning that before 1990, our model assigned East Germany to the same group as other eastern European countries and USSR (Russia), while it assigned West Germany to the same group as western European countries.[2]

[1] http://en.wikipedia.org/wiki/List_of_countries_by_GDP_%28PPP%29. In Table 8, we omit some countries (represented by ...) in order to incorporate other interesting but relatively low ranked countries (for example, Russia) in the GDP list.

[2] Not shown in Table 9 because missing from 2005 GDP data.

Time Period	Topic 1	Topic 2	Topic 3	Group distributions for Topic 3				
				Group 1	Group2	Group3	Group4	Group5
60-75	Nuclear	Procedure	Africa Indep.	India	USA	Argentina	USSR	Turkey
	operative	committee	calling	Indonesia	Japan	Colombia	Poland	
	general	amendment	right	Iran	UK	Chile	Hungary	
	nuclear	assembly	africa	Thailand	France	Venezuela	Bulgaria	
	power	deciding	self	Philippines	Italy	Dominican	Belarus	
65-80	Independence	Finance	Weapons	Cuba	India	Algeria	USSR	USA
	territories	budget	nuclear	Albania	Indonesia	Iraq	Poland	Japan
	independence	appropriation	UN		Pakistan	Syria	Hungary	UK
	self	contribution	international		Saudi	Libya	Bulgaria	France
	colonial	income	weapons		Egypt	Afghanistan	Belarus	Italy
70-85	N. Weapons	Israel	Rights	Mexico	China	USA	Brazil	India
	nuclear	israel	africa	Indonesia		Japan	Turkey	USSR
	international	measures	territories	Iran		UK	Argentina	Poland
	UN	hebron	south	Thailand		France	Colombia	Vietnam
	human	expelling	right	Philippines		Italy	Chile	Hungary
75-90	Rights	Israel/Pal.	Disarmament	Mexico	USA	Algeria	China	India
	south	israel	UN	Indonesia	Japan	Vietnam	Brazil	
	africa	arab	international	Iran	UK	Iraq	Argentina	
	israel	occupied	nuclear	Thailand	France	Syria	Colombia	
	rights	palestine	disarmament	Philippines	USSR	Libya	Chile	
80-95	Disarmament	Conflict	Pal. Rights	USA	China	Japan	Guatemala	Malawi
	nuclear	need	rights	Israel	India	UK	St Vincent	
	US	israel	palestine		Russia	France	Dominican	
	disarmament	palestine	israel		Spain	Italy		
	international	secretary	occupied		Hungary	Canada		
85-00	Weapons	Rights	Israel/Pal.	Poland	China	USA	Russia	Cameroon
	nuclear	rights	israeli	Czech R.	India	Japan	Argentina	Congo
	weapons	human	palestine	Hungary	Brazil	UK	Ukraine	Ivory C.
	use	fundamental	occupied	Bulgaria	Mexico	France	Belarus	Liberia
	international	freedoms	disarmament	Albania	Indonesia	Italy	Malta	

Table 9: Results for 15-year-span slices of the UN dataset (1960-2000). The top probable words are listed for all topics, but only the groups corresponding the most dominant topic are shown (Topic 3). We list the countries for each group ordered by their 2005 GDP (PPP)and only show the top 5 countries in groups that have more than 5 members. We do not repeat the results in Table 8 for the most recent window (1990-2003).

5. CONCLUSIONS

We present the Group-Topic model that jointly discovers latent groups in a network as well as clusters of attributes (or topics) of events that influence the interaction between entities in the network. The model extends prior work on latent group discovery by capturing not only pair-wise relations between entities but also multiple attributes of the relations (in particular, the model considers words describing the relations). In this way the GT model obtains more cohesive groups as well as fresh topics that influence the interaction between groups. The model could be applied to variables of other data types in addition to voting data. We are now using the model to analyze the citations in academic papers to capture the topics of research papers and discover research groups. It would also apply to a much larger network of entities (people, organizations, etc.) that frequently appear in newswire articles.

The model can be altered suitably to consider other attributes characterizing relations between entities in a network. In ongoing work we are extending the Group-Topic model to capture a richer notion of topic, where the attributes describing the relations between entities are represented by a mixture of topics.

6. ACKNOWLEDGMENTS

This work was supported in part by NSF grant #IIS-0326249, and by the Defense Advanced Research Projects Agency, through the Department of the Interior, NBC, Acquisition Services Division, under contract #NBCHD030010. We would also like to greatly thank Prof. Vincent Moscardelli, Chris Pal and Aron Culotta for helpful discussion.

7. REFERENCES

[1] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *SIGKDD*, 2000.

[2] R. Bekkerman, R. E. Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *ICML*, 2005.

[3] I. Bhattacharya and L. Getoor. Deduplication and group detection using links. In *LinkKDD*, 2004.

[4] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] K. Carley. A theory of group stability. *American Sociological Review*, 56(3):331–354, 1991.

[6] K. Carley. A comparison of artificial and human organizations. *Journal of Economic Behavior and Organization*, 56:175–191, 1996.

[7] G. Cox and K. Poole. On measuring the partisanship in roll-call voting: The U.S. House of Represenatatives, 1887-1999. *American Journal of Political Science*, 46(1):477–489, 2002.

[8] W. W. Denham, C. K. McDaniel, and J. R. Atkins. Aranda and Alyawarra kinship : A quantitative argument for a double helix model. *American Ethnologist*, 6(1):1–24, 1979.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *SIGKDD*, 2003.

[10] D. Fenn, O. Suleman, J. Efstathiou, and N. Johnson. How does Europe make its mind up? Connections, cliques, and compatibility between countries in the Eurovision song contest. *arXiv:physics/0505071*, 2005.

[11] S. Hix, A. Noury, and G. Roland. Power to the parties: Cohesion and competition in the European Parliament, 1979-2001. *British Journal of Political Science*, 35(2):209–234, 2005.

[12] A. Jakulin and W. Buntine. Analyzing the US Senate in 2003: Similarities, networks, clusters and blocs, 2004.

[13] C. Kemp, T. L. Griffiths, and J. Tenenbaum. Discovering latent classes in relational data. Technical report, MIT CSAIL, 2004.

[14] D. Krackhardt and K. M. Carley. A PCANS model of structure in organization. In *Int. Sym. on Command and Control Research and Technology*, June 1998.

[15] J. Kubica, A. Moore, J. Schneider, and Y. Yang. Stochastic link and group detection. In *AAAI*, 2002.

[16] A. McCallum, A. Corrada-Emanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI*, 2005.

[17] K. Nowicki and T. A. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455), 2001.

[18] A. Pajala, A. Jakulin, and W. Buntine. Parliamentary group and individual voting behavior in Finnish Parliamentin year 2003 : A group cohesion and voting similarity analysis, 2004.

[19] M. Sparrow. The application of network analysis to criminal intelligence: an assessment of prospects. *Social Networks*, 13:251–274, 1991.

[20] E. Voeten. Documenting votes in the UN General Assembly. http://home.gwu.edu/∼voeten/UNVoting.htm_Toc82404232.

[21] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

APPENDIX

A. GIBBS SAMPLING DERIVATIONS

Begin with the joint distribution $P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)$, we can take the advantages of conjugate priors to simplify the formulae. All symbols are defined in Sec. 2.

$$
\begin{aligned}
& P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta) \\
= & \iiint p(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}, \theta, \gamma, \phi|\alpha, \beta, \eta)\mathrm{d}\theta\mathrm{d}\gamma\mathrm{d}\phi \\
= & \iiint \prod_{b=1}^{B} P(t_b) \prod_{t=1}^{T} \left(p(\theta_t|\alpha) \prod_{s=1}^{S} P(g_{st}|\theta_t) p(\phi_t|\eta) \right) \\
& \times \prod_{b=1}^{B} \prod_{g=1}^{G} \prod_{h=g}^{G} p(\gamma_{gh}^{(b)}|\beta) \prod_{b=1}^{B} \prod_{i=1}^{N_b} P(w_i^{(b)}|\phi_{t_b}) \\
& \times \prod_{b=1}^{B} \prod_{i=1}^{S} \prod_{j=i+1}^{S} P(V_{ij}^{(b)}|\gamma_{g_i g_j}^{(b)}))\mathrm{d}\theta\mathrm{d}\gamma\mathrm{d}\phi \\
= & \iiint \left(\frac{1}{T}\right)^{B} \prod_{t=1}^{T} \left(\frac{\Gamma(\sum_{g=1}^{G} \alpha_g)}{\prod_{g=1}^{G} \Gamma(\alpha_g)} \prod_{g=1}^{G} \theta_{tg}^{\alpha_g - 1} \prod_{g=1}^{G} \theta_{tg}^{n_{tg}} \right) \\
& \times \prod_{t=1}^{T} \left(\frac{\Gamma(\sum_{v=1}^{V} \eta_v)}{\prod_{v=1}^{V} \Gamma(\eta_v)} \prod_{v=1}^{V} \phi_{tv}^{\eta_v - 1} \right) \\
& \times \prod_{b=1}^{B} \prod_{g=1}^{G} \prod_{h=g}^{G} \left(\frac{\Gamma(\sum_{k=1}^{2} \beta_k)}{\prod_{k=1}^{2} \Gamma(\beta_k)} \prod_{k=1}^{2} (\gamma_{ghk}^{(b)})^{\beta_k - 1} \right) \\
& \times \prod_{t=1}^{T} \prod_{v=1}^{V} \phi_{tv}^{c_{tv}} \prod_{b=1}^{B} \prod_{g=1}^{G} \prod_{h=g}^{G} \prod_{k=1}^{2} (\gamma_{ghk}^{(b)})^{m_{ghk}^{(b)}} \mathrm{d}\theta\mathrm{d}\gamma\mathrm{d}\phi \\
\propto & \iiint \prod_{t=1}^{T} \prod_{g=1}^{G} \theta_{tg}^{\alpha_g + n_{tg} - 1} \prod_{t=1}^{T} \prod_{v=1}^{V} \phi_{tv}^{\eta_v + c_{tv} - 1} \\
& \times \prod_{b=1}^{B} \prod_{g=1}^{G} \prod_{h=g}^{G} \prod_{k=1}^{2} (\gamma_{ghk}^{(b)})^{\beta_k + m_{ghk}^{(b)} - 1} \mathrm{d}\theta\mathrm{d}\gamma\mathrm{d}\phi \\
\propto & \prod_{t=1}^{T} \left(\frac{\prod_{g=1}^{G} \Gamma(\alpha_g + n_{tg})}{\Gamma(\sum_{g=1}^{G} (\alpha_g + n_{tg}))} \frac{\prod_{v=1}^{V} \Gamma(\eta_v + c_{tv})}{\Gamma(\sum_{v=1}^{V} (\eta_v + c_{tv}))} \right) \\
& \times \prod_{b=1}^{B} \prod_{g=1}^{G} \prod_{h=g}^{G} \frac{\prod_{k=1}^{2} \Gamma(\beta_k + m_{ghk}^{(b)})}{\Gamma(\sum_{k=1}^{2} (\beta_k + m_{ghk}^{(b)}))}
\end{aligned}
$$

Using the chain rule, we can get the conditional probability conveniently,

$$
\begin{aligned}
& P(g_{st}|\mathbf{V}, \mathbf{g}_{-st}, \mathbf{w}, \mathbf{t}, \alpha, \beta, \eta) \\
= & \frac{P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)}{P(\mathbf{g}_{-st}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)} \\
\propto & \frac{P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)}{P(\mathbf{g}_{-st}, \mathbf{V}_{-st}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)} \\
\propto & \frac{\alpha_{g_{st}} + n_{tg_{st}} - 1}{\sum_{g=1}^{G} (\alpha_g + n_{tg}) - 1} \prod_{b=1}^{B} \left(I(t_b = t) \right. \\
& \left. \times \prod_{h=1}^{G} \frac{\prod_{k=1}^{2} \prod_{x=1}^{d_{g_{st}hk}^{(b)}} \left(\beta_k + m_{g_{st}hk}^{(b)} - x\right)}{\prod_{x=1}^{\sum_{k=1}^{2} d_{g_{st}hk}^{(b)}} \left((\sum_{k=1}^{2} (\beta_k + m_{g_{st}hk}^{(b)}) - x\right)} \right)
\end{aligned}
$$

and,

$$
\begin{aligned}
& P(t_b|\mathbf{V}, \mathbf{g}, \mathbf{w}, \mathbf{t}_{-b}, \alpha, \beta, \eta) \\
= & \frac{P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)}{P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}_{-b}|\alpha, \beta, \eta)} \\
\propto & \frac{P(\mathbf{g}, \mathbf{V}, \mathbf{w}, \mathbf{t}|\alpha, \beta, \eta)}{P(\mathbf{g}, \mathbf{V}_{-b}, \mathbf{w}_{-b}, \mathbf{t}_{-b}|\alpha, \beta, \eta)} \\
\propto & \frac{\prod_{v=1}^{V} \prod_{x=1}^{e_v^{(b)}} (\eta_v + c_{t_b v} - x)}{\prod_{x=1}^{\sum_{v=1}^{V} e_v^{(b)}} \left(\sum_{v=1}^{V} (\eta_v + c_{t_b v}) - x\right)} \\
& \times \prod_{g=1}^{G} \prod_{h=g}^{G} \frac{\prod_{k=1}^{2} \Gamma(\beta_k + m_{ghk}^{(b)})}{\Gamma(\sum_{k=1}^{2} (\beta_k + m_{ghk}^{(b)}))}
\end{aligned}
$$