Bulletin of the Technical Committee on

Data Engineering

June 2007 Vol. 30 No. 2

IEEE Computer Society

Letters

Special Issue on Data Management Issues in Social Sciences

Conference and Journal Notices

The Bulletin of the Technical Committee on Data Engineering is published quarterly and is distributed to all TC members. Its scope includes the design, implementation, modelling, theory and application of database systems and their technology.

Letters, conference information, and news should be sent to the Editor-in-Chief. Papers for each issue are solicited by and should be sent to the Associate Editor responsible for the issue.

Opinions expressed in contributions are those of the authors and do not necessarily reflect the positions of the TC on Data Engineering, the IEEE Computer Society, or the authors' organizations.

Membership in the TC on Data Engineering is open to all current members of the IEEE Computer Society who are interested in database systems.

There are two Data Engineering Bulletin web sites: http://www.research.microsoft.com/research/db/debull and http://sites.computer.org/debull/.
The TC on Data Engineering web page is http://www.ipsi.fraunhofer.de/tcde/.

Letter from the Editor-in-Chief

Executive Committee of the TCDE

The inside front cover of this issue of the Bulletin includes the new Executive Committee of the IEEE Technical Committee on Data Engineering, the sponsoring organization within the IEEE for database activities, including the Bulletin and the ICDE Conference. Matters pertaining to the TCDE should be addressed to Paul Larson, the TCDE Chair.

Workshop on Self-Managing Database Systems

The current issue contains a report on the second Workshop on Self-Managing Database Systems, sponsored by the Workgroup on Self-Managing Data, a workgroup of the IEEE Technical Committee on Data Engineering. This workgroup was formed less than two years ago and has already sponsored two workshops and had an issue of the Bulletin devoted to the topic. Self-managing database systems is a subject of keen industrial interest as users focus on reducing total cost of ownership (TCO) for their data processing systems. The workshop report on page three captures some of the real excitement in this area.

The Current Issue

The current issue of the Bulletin is on the topic of data management issues in social sciences. Computers and the internet are having a striking impact of the lives of not only technical communities but on communities of what might be called ”ordinary” users. Web sites cater to social interaction, becoming places to ”meet”, socialize, post videos, etc. Web search enables people distributed physically over much of the earth's surface to find likeminded individuals who share their interests. This is has become a really dramatic phenomenon.

The database community brings to bear on social science issues technologies that it has developed over the years for managing business data. But the social sciences require new ways of looking at and ”massaging” data. This has led to ideas and constructs such as social and affiliation networks, new search paradigms, and new ways to organize systems to support these technologies. We can see this social phenomenon changing in real time, and hence it presents a moving target, as social scientists struggle to keep up with what is happening.

I want to thank issue editor Nick Koudas (and Dimitris Tsirogiannis who provided editorial assistence), who has brought together in the current issue a sampling of the substantial amount of work going on in the social sciences by folks who are in or close to the database community. There is real excitement in this area, including industrial excitement as companies try to figure out the best way to attract users to their web site– which is one key to attracting advertisers. The business model for the web is now pretty clearly advertisement based, so you can expect this kind of ”social science” work to remain an important area for many years to come.

David Lomet
Microsoft Corporation

Report on the Second International Workshop on Self-Managing Database Systems (SMDB 2007)

Anastassia Ailamaki
Carnegie Mellon University
natassa@cmu.edu

Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com

Sam Lightstone
IBM Software Development Laboratory
light@ca.ibm.com

Guy Lohman
IBM Almaden Research Center
lohman@almaden.ibm.com

Pat Martin
Queen's University
martin@cs.queensu.ca

Ken Salem
University of Waterloo
kmsalem@uwaterloo.ca

Gerhard Weikum
Max-Planck Institut fr Informatik
weikum@mpi-sb.mpg.de

1 Introduction

Information management systems are growing rapidly in scale and complexity, while skilled database administrators are becoming rarer and more expensive. Increasingly, the total cost of ownership of information management systems is dominated by the cost of people, rather than hardware or software costs. This economic dynamic dictates that information systems of the future be more automated and simpler to use, with most administration tasks transparent to the user.

Autonomic, or self-managing, systems are a promising approach to achieving the goal of systems that are increasingly automated and easier to use. The aim of the workshop was to provide a forum for researchers from both industry and academia to present and discuss ideas related to self-managing database systems.

SMDB 2007 was the first event organized by the new **IEEE Computer Society Data Engineering Workgroup on Self-Managing Database Systems** ($http : //db.uwaterloo.ca/tcde - smdb/$). The workgroup, which was founded in October 2005, is intended to foster research aimed at enabling information management systems to manage themselves seamlessly, thereby reducing the cost of deployment and administration.

2 Workshop Overview

The workshop was held in Istanbul, Turkey on Monday April 16, 2007 prior to the start of the International Conference on Data Engineering. The workshop's program committee consisted of the members of the SMDB workgroup executive committee plus four other well-known researchers in the area. SMDB 2007 received 19 submissions and each paper was reviewed by 3 program committee members. 11 papers were accepted to the workshop, resulting in an acceptance rate of 58%. In an effort to make the workshop as inclusive as possible 4

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

submissions were accepted as poster papers and were given a shorter presentation time in the workshop. The average attendance at the workshop during the day was 32 participants.

3 Technical Program

The technical program was organized into four sessions: principles and overviews; self-healing; self-optimizing and poster papers. Links to the talks and papers can be found at the workshop Web page ($http : //db.uwaterloo.ca/tcde - smdb/SMDB2007_Program.html$).

The first session on principles and overviews included three research papers highlighting key research issues in self-managing database systems. Soror et al. [1] considered the impact that the trend to virtualization will have on tuning database management systems (DBMSs). They presented a formalization of the resource allocation problem in this environment and discussed an approach to cost modeling that employs a query optimizer with a "what-if" mode. Lightstone et al. [2] argued for a greater use of control theory in self-managing database systems and presented examples in IBM's DB2 where it was used effectively to control utility throttling and self-managing memory. They found the main advantages of control theory are its stability and its ability to handle noise. Chen et al. [3] observed that the currently available coarse-grained resource provisioning solutions do not necessarily make effective use of the available resources. They proposed a fine-grained approach that uses outlier detection to pinpoint sources of overload problems and then migrates these queries. The fourth paper in the session, by Bowman et al. [4], provided an overview of the self-management features of SQL-Anywhere from Sybase iAnywhere. SQL-Anywhere is designed to be deployed as an embedded DBMS within zero-administration environments.

The second session, which was on self-healing, included two papers. Cook et al. [5] defined the general problem of supporting self-healing in database-centric multi-tier services and outlined a research agenda for solving the problem. They specifically identified performance availability problems as reasonable targets for self-healing and supported an approach based on robust learning algorithms. Modani et al. [6] described an approach to automated diagnosis in which symptoms of a new problem are matched to a database of symptoms of previously diagnosed problems. They exploited the fact that function call stacks can serve as symptoms of a class of problems and proposed algorithms for effectively matching call stack patterns.

The third session of the workshop included four papers on self-tuning. Papadomanolakis and Ailamaki [7] observed that existing index selection tools rely on heuristics that are hard to analyze. They proposed a model for index selection based on integer linear programming that offers higher solution quality, efficiency and scalability without sacrificing any of the precision offered by existing index selection tools. Sattler et al. [8] examined a different aspect of the index selection problem. They proposed an approach that continuously collects statistics for recommended indexes and performs on-the-fly index generation during query processing using new query plan operators IndexBuildScan and SwitchPlan. Schnaitter et al. [9] also considered an aspect of automatic index selection, namely the selection of indexes as the workload on a DBMS shifts characteristics. They described COLT, which is a novel framework that continuously monitors the workload of a database system and enriches the existing physical design with a set of effective indices. Qin et al. [10] looked to improve the cost models employed by query optimizers by improving the accuracies of I/O cost estimates of database access methods. They presented an adaptive black box statistical cost estimation method.

The fourth session of the workshop included four poster papers on a variety of topics in the area of self-managing database systems. The poster papers were each given a shorter presentation time than the regular papers. Niu et al. [11] described an approach to automatically adapting DBMS workloads such that service level objectives of the various applications are met. Duchateau et al. [12] presented an automatic schema matching approach. They specifically focused on a B-tree index structure to improve the performance of the matching algorithm. Teisanu et al. [13] outlined the problem of designing effective workload management and provided a formal definition that supports the further development of algorithms and architectures for effective on-line database tuning strategies. Lang et al. [14] described a caching algorithm for scans on buffer pools that keeps track of ongoing scans and the state of each scan. They showed that this approach could achieve improved buffer pool hit rates.

4 Summary

The Second Workshop on Self-Managing Database Systems was very successful. The high quality of the papers and the discussions generated during the workshop are strong indicators of the vitality and growing importance of the area of self-managing information management systems.

The Workgroup on Self-Managing Database Systems looks forward to organizing the third edition of the workshop along with ICDE 2008 in Cancun. They seek to encourage a wider range of submissions and a broader participation by academics and industrial partners in the area.

References

[1] A. Soror, A. Aboulnaga, and K. Salem, "Database virtualization: A new frontier for database tuning and physical design," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 388 – 393, Istanbul Turkey, 2007.

[2] S. Lightstone, M. Surendra, Y. Diao, S. Parekh, J. Hellerstein, K. Rose, A. Storm, and C. Garcia-Arellano, "Control theory: a foundational technique for self managing databases," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 394 – 403, Istanbul Turkey, 2007.

[3] J. Chen, G. Soundararajan, M. Mihailescu, and C. Amza, "Outlier detection for fine-grained load balancing in database clusters," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 404 – 413, Istanbul Turkey, 2007.

[4] I. Bowman, P. Bumbulis, D. Farrar, A. Goel, B. Lucier, A. Nica, G. Paulley, J. Smirnios, and M. Young-Lai, "Sql anywhere: A holistic approach to database self-management," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 414 – 423, Istanbul Turkey, 2007.

[5] B. Cook, S. Babu, G. Candea, and S. Duan, "Toward self-healing multitier services," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 424 – 432, Istanbul Turkey, 2007.

[6] N. Modani, R. Gupta, G. Lohman, T. Syeda-Mahmood, and L. Mignet, "Automatically identifying known software problems," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 433 – 441, Istanbul Turkey, 2007.

[7] S. Papadomanolakis and A. Ailamaki, "An integer linear programming approach to database design," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 442 – 449, Istanbul Turkey, 2007.

[8] K.-U. Sattler, M. Luehring, K. Schmidt, and E. Schallehn, "Autonomous management of soft indexes," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 450 – 458, Istanbul Turkey, 2007.

[9] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis, "On-line index selection for shifting workloads," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 459 – 468, Istanbul Turkey, 2007.

[10] Y. Qin, K. Salem, and A. Goel, "Towards adaptive costing of database access methods," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 469 – 477, Istanbul Turkey, 2007.

[11] B. Niu, P. Martin, W. Powley, P. Bird, and R. Horman, "Poster session: Adapting mixed workloads to meet slos in autonomic dbmss," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 478 – 484, Istanbul Turkey, 2007.

[12] F. Duchateau, Z. Bellahsene, M. Roantree, and M. Roche, "Poster session: An indexing structure for automatic schema matching," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 485 – 491, Istanbul Turkey, 2007.

[13] A. Teisanu, M. Consens, M. Kandil, S. Lightstone, D. Zilio, and C. Zuzarte, "Poster session: Problem definition for effective workload management," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 492 – 497, Istanbul Turkey, 2007.

[14] C. Lang, B. Bhattacharjee, T. Malkemus, and I. Stanoi, "Poster session: Improved buffer size adaptation through cache/controller coupling," in *Proceedings of ICDE Workshops (SMDB 2007)*, pp. 498 – 506, Istanbul Turkey, 2007.

Letter from the Special Issue Editor

The field of social network analysis has been an active research area in social sciences for a long time. In the last couple of years primarily via the manifestation of social networks of large scale through web applications the field has enjoyed active research participation from multiple communities. Recent applications such as social networking sites (MySpace, LinkedIn, Facebook, to name a few), media sharing and collaboration sites (e.g., Flickr, YouTube), blog applications (e.g., Blogger, LiveJournal, NotePad) gave rise to structured and adhoc social networks of massive scale. Social scientists, physicists, computer scientists and engineers are trying to study and understand the properties of these networks and the challenges they pose.

Distinguishing characteristics of these networks to those studied before is their massive scale and their dynamic nature. Such characteristics pose certain challenges to handle the data volume and their dynamics and need to be carefully understood. Moreover, they enable new applications and raise research challenges both at the modeling and algorithmic level. Data management has a lot to offer in terms of addressing such challenges.

The purpose of this volume is twofold. First, to collect and present works that highlight some of the research challenges lying ahead that our community started to address. Towards this end we have collected articles listing challenges in information management in a social networks context (articles by AnHai Doan et. al., and Amer-Yahia et. al.) as well as articles demonstrating interesting problems and techniques resulting from the structure implicit in such networks (articles by Singh and Getoor and Bender et. al.,). The article by Adar and Re presents an interesting connection between problems in the social networking area and the recent work on probabilistic data management. The second purpose of this volume is to bring the perspective of social scientists that have been working on social network analysis to the table. The article by Bernie Hogan provides a brief overview to social network analysis from the social sciences perspective and presents the data management problems part of this community faces when dealing with social networks of large scale.

The research challenges lying ahead are important and likely to become more significant as the web evolves to a global dynamic and interactive collective. It is important to remember that we are not into this alone, other communities are contributing to this effort and we need to foster interaction among communities and exchange of research ideas. I sincerely hope that this volume contributes towards this direction.

I wish to thank Mr. Dimitris Tsirogiannis from the University of Toronto for editorial assistance with this volume.

Nick Koudas
University of Toronto
Toronto, Canada

Using Information Networks to Study Social Behavior: An Appraisal

Bernie Hogan

Abstract

Social network analysis investigates relationships between people and information created by people. The field is presently in flux due to the increasing amount of available data and the concomitant interest in networks across many disciplines. This article reviews some of the recent advances in the field, such as p modeling and community detection algorithms alongside some of the societal transitions that facilitate these advances. The latter third of the article raises some issues for data engineers to consider given the state of the field. These issues focus mainly on querying and managing large and complex datasets, such as those commonly found through online scraping.*

1 Introduction

1.1 Social Networks and Digital Traces

This is a particularly exciting time to be a social network researcher; advances both within social network analysis and within society at large are making our work increasingly relevant. To be clear, by social networks we refer to associations between humans, or information created by humans. Presently, widespread use of information and communication technologies (ICTs) such as email, social software and cell phones have made possible the creation, distribution and aggregation of these relationships on an entirely different scale. But as we broaden our reach and seek ever more sophisticated answers within this paradigm, it is clear that we cannot do it alone. Quite frankly, there is much work to do and a substantial chunk of this work can get very technical, very fast. This article presents a brief overview of social network analysis as a field, where it is heading given current advances in the field, and where it may head as social scientists forge stronger links with computer scientists. As Gray and Szalay [1] note about science in general, there is now a data avalanche in the social sciences, and despite much of our previous expectations, we have indeed become data rich.

The use of digital media means that information can be copied and aggregated for analysis with very little extra work on the part of the respondent. Prior to the proliferation of digital media, gathering data about relations between ties was a long and expensive affair. Most data was self-reported, meaning an interviewer had to ask each respondent in turn. Not only did this place practical limits on the size of populations, but it introduced methodological issues of recall bias and concordance [2][3][1]. Some researchers have persuasively argued that individuals are good at recalling trends over time [4], but this is still not as robust as passive traces.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1]Concordance refers to the extent to which two individuals will report a relationship of similar strength or frequency.

Digital traces are particularly significant artifacts for researchers. With email, for example, we are not left with empty postmarked envelopes or a caller ID from a telephone call - we are left with the names, dates, content and recipients. And because this interaction is digital, there is virtually no marginal cost in making a perfect replica of the messages for analysis. This level of fidelity and efficiency means that studies of social activity can (and have) scaled from previous 'huge' data sets of a thousand people, to a data set of millions of messages. The consequence of this is to shift the burden from the practicality of collecting the data to the practicality of managing it.

These digital traces are probably more significant for social network studies than traditional social science methods. This is because much network analysis does not deal with sampling very well. The absence of a few key ties could completely alter the profile of a network [5][6]. While the probability that one would miss a key tie is low, their absence is significant. This confined network analysis to small whole groups, such as an office team or corporate board (with cases numbering from 10 up to a few hundred) or samples of 'personal networks' where a person reports on the *perceived* ties between friends and family. Now, however, complete social networks such as the world wide web, or massive corporate communication networks are available. Even snowball sampling, which in face-to-face methods is costly and slow, can now be done simply by following hyperlinks from MySpace pages or a weblog (blog).

We can now scrape company inboxes and map the entire communication network for a large scale firm [7], plot links between ideologically charged blog discussions [8], or even map the entire email network for a university domain [9]. Asking all of the participants for this information directly is still possible, but non-obtrusive measures are so comprehensive and often efficient that many researchers start with trace data rather than consider it mere supplement. This situation has brought with it the usual issues about data quality (i.e. how do we capture this data, what data is worth keeping and what should be discarded), but social scientists deal with people, not star clusters or genetic maps. This leads to a host of additional questions that we are only now learning how to ask with trace data, let alone answer. Issues of data aggregation, privacy, contamination (e.g. the "Hawthorne Effect"[2]), partial/incomplete data sources and ethical approval are relevant when dealing with human subjects. Moreover, many of these issues are located at the database level, where strict policies (such as one-way salted hashing of names) come into play and represent compromises between researchers and subjects. Finally, there are complex ontological issues that must be addressed - is a Facebook friend really an important friend [11]? What sort of additional data must be captured to distinguish 'friendsters' from friends?

The remaining bulk of this article will present an overview of various advances in social network analysis, and exogenous advances that affect social network analysis. Throughout, issues of data management will either be in focus, or at least in the peripheral vision. Before concluding, I will also introduce some recent practical issues at the nexus of data engineering and social science methodologies.

2 A brief history of social network analysis

2.1 Early years

As a paradigm, network analysis began to mature in the 1970s. In 1969, Stanley Milgram published his Small World experiment, demonstrating the now colloquial "six degrees of separation"[12]. In 1973, Mark Granovetter published the landmark "The Strength of Weak Ties" which showed empirically and theoretically how the logic of relationship formation led to clusters of individuals with common knowledge and important 'weak tie' links between these clusters [13]. As he showed, job searchers are most successful not by looking to their strong ties (who have similar information as the searcher) but to their weak ties who link to other pockets of information. This decade also saw the first major personal network studies [14][15], an early, but definitive, statement on

[2]This effect refers to any situation where a general stimulus alters the subject's behavior. It is particularly problematic because it is not a specific stimulus that creates a change, but mere attention[10]

network metrics [16], and the formation of two journals (Social Networks and Connections) and an academic society (The International Network of Social Network Analysts). The following two decades saw explosive growth in the number of studies that either alluded to or directly employed network analysis. This includes work on the interconnectedness of corporate boards[19], the core discussion networks of Americans [20], the logic of diffusion,whether it's the diffusion of the latest music format or the spread of a new disease[21],and even the social structure of nation states [22].

2.2 The last decade

Increasing computational power and the dawn of the Internet ushered in the second major shift in network thinking. By this point, physicists, biologists, and information scientists started contributing to a larger paradigm of 'network science'. Massive data sets could be gathered and analyzed in reasonable time frames, leading to maps and insights not only about a schoolyard or a few hundred personal networks, but about the billions of nodes on the World Wide Web. This era produced advances which I will categorize in three sections: Endogenous advances - namely those advances coming from the field of social network analysis proper, parallel advances - those coming from related fields and scientific endeavors and exogenous advances - those coming from society outside academia but having a significant effect on the field.

3 Endogenous advances

As was mentioned above, up until recent years, network analysis has traditionally been concerned with relatively small data sets or sampled populations. For example, numerous tutorials have been given on "Krackhart's High tech managers", a study of a whole network with a mere 21 respondents (see [23][24]). These specialized populations have led to analytic tools for such small networks. In general, the techniques either focus on relations, positions or motifs.

3.1 Relations

Relations are perhaps the most intuitive concept in social network analysis. Simply stated, these are methods for describing the importance of direct links between nodes in the network. These generally fall into two categories: examinations of the most prominent individuals and examinations of subgroups or communities of individuals. For measures of prominence, most research continues to use classic centrality measures (degree, closeness and betweenness centrality), even though two of them are computationally expensive. Specifically, betweenness centrality and closeness centrality both look for the shortest path between two nodes [16]. Since the creation of these measures a few other notable measures have cropped up for evaluating the prominence of an individual. These include eigenvector centrality which weights a nodes centrality score by the score of one's neighbors (thus one might be considered central not because they have many ties, but ties to popular people) [17]. While not a social network method, Google's PageRank is similar to Eigenvector centrality except it doesn't require complete information about the network [18]. Recent advances in this area have paid more attention to the robustness of these measures under varying conditions, than to the elaboration of new measures [5][22][6].

The relational analysis of subgroups is a field of active research. Here one looks at the ties between nodes to come up with subgroups or 'communities'. Early sociological work looked at the analysis of cliques (or complete subgraphs) whereas computer science solutions examined network flows. Questions about automatic detection of community structure are becoming increasingly relevant, as sites seek to categorize the structure of their users. In recent years, methods have moved beyond various max-flow min-cut solutions towards methods based on expected density and edge betweenness [25][26].

Figure 1: Example network - represented as a sociogram and a matrix. The right hand side shows the reduced matrix as a result of blocking the network.

3.2 Positions

Positions are a more lateral concept, stemming primarily from the work of Harrison White and his contemporaries in the early 1970s. If relations are concerned about *who* one is connected to, then positions are concerned with *how* individuals are connected. For example, the world system has a core-periphery structure, with most international finance moving through a few key cities. Those cities on the periphery might not be connected to each other, but they might be connected to the main cities in the same manner - hence they have similar positions in the system of global trade. Two nodes are considered structurally equivalent if they are connected to the same specific nodes in the same way. Two nodes are considered to be regularly equivalent if they are connected to any nodes in the same way. Partitioning a network into equivalent sets is referred to blockmodeling [27].

Figure 1 shows how nodes A and B are connected in equivalent ways, as are nodes D and E. Once the graph is partitioned using a blockmodeling algorithm, one can reduce the graph to its clusters, as is seen in the right-hand side of figure 1. To note, when you reduce a graph to its clusters you can have self-loops as nodes within each cluster can link to other nodes in that cluster.

As a technique, blockmodeling has a number of distinct strengths. Most particularly, this technique can find clusters of relationships which might otherwise be hidden. Community detection algorithms generally base their insights on higher connectivity within a subgraph than between subgraphs. Blocks, by contrast focus on the pattern of connectivity rather than the prevalence of connections. One non-sociological example of this is the work of Broder et al. [28] in characterizing the web based on positions (this includes a strongly connected core, an in-group, an out-group, tendrils, tubes and islands).

The last decade has seen two major improvements in blockmodeling. The first is generalized blockmodeling, enabling partitions of smaller, more precise block types, counts of 'errors' for fitting models and predefined block structures based on attribute data[29]. The second is stochastic blockmodeling which compares a partition to partitions from similar networks to attain a probabilistic goodness-of-fit statistic [30]. In both cases, there are still a number of open questions. One is how to interpret partitions of massive networks, when it is unclear what will constitute an optimally fitting partition. The second is what to do with various edge types. When we consider an edge valued as 1 or 0, partitioning is straightforward, but blocking on data that is signed (positive, neutral or negative ties), sequential or weighted is still very experimental.

3.3 Motifs / Configurations

Motifs are small easily defined network structures which can be combined to create larger networks [31]. In social network analysis these are generally called configurations and represented in p* / Exponential Random Graph models [32]. There are only 3 dyadic configurations between two nodes (a symmetric tie, an asymmetric tie and no tie), but numerous configurations between the sixteen possible triadic motifs[33][30]. There are numerous theoretical reasons to believe that these configurations can be interpreted meaningfully, and that their distribution can inform the processes of tie formation and decay that characterize network dynamics [34].

Exponential Random Graph models refer to a family of probability distributions used to assess the likelihood of a particular network configurations appearing by chance. Testing these models is often done using either Maximum Pseudolikelihood Estimation or Monte Carlo Markov Chain Maximum Likelihood Estimation. While the former is far more efficient, it is often too conservative with standard errors and certain distributions and therefore should only be considered a proximate tool (Wasserman and Robins 2005). The latter is so computationally expensive that some models may not converge after days of iterations. Nevertheless, a robust model can illustrate quite clearly the relative importance of certain micro structures (such as 2-stars) on the emergent macro structures. Some of the most significant open problems in this area are related to the optimization of these methods and the use of robust estimation techniques. Because of the complexity of these dependency structures (e.g. requiring so many triads or four-cycles), and the fact that many of these problems appear to be NP-complete, advances in both computational algorithms and storage are welcome additions to the field.

4 Parallel Advances

4.1 Physicists, Biologists and Computer Scientists, Oh My!

Presently, there are far more individuals working in network data than social scientists and mathematicians. Biologists, physicists, are computer scientists are among the many disciplines that are finding network research particularly relevant to many of their research questions. Take the web, for example. It was created by humans and its linking structure is the result of many individual decisions. Yet, physicists have been characterizing the structure of the web as an emerging from many small decisions. In this vein, Watts and Strogatz showed that Milgram's small worlds (which they formally characterized as networks with high clustering and short paths between actors) could be found in movie actor networks and neural structures alike [35]. Through an analysis of virtually the entire World Wide Web, Barabasi and Albert [36] illustrated a major class of networks known as "scale-free networks", which have been subsequently found in traffic patterns, DNA and online participation [37]. All of these scale-free networks are based on the incredibly straightforward logic of preferential attachment: as a network grows, nodes with more links are likely to attract even more links, thus creating massive asymmetries between the few nodes with many links and the many nodes with few.

Biologists are finding that the human genome is an incredibly efficient means of encoding the data necessary for life. Genes do not work like on-off switches for direct effects, but work in combination, such that if two or more genes are present there is a particular phenotypical expression, but without all of them, there is none. This will have great consequences in the understanding of genetic diseases, as certain diseases depend on particular genes - but - other parts of the genome also depend on these focal genes.

As is probably evident to this audience, the use of network models in computer science has led to a number of very efficient solutions to problems. The 500-pound gorilla in the room is no doubt Google, who have used PageRank (and certainly a modified version thereof) to make search results more credible. Google succeeded as many people found their solution more useful than other patterns based on keywords or categories.

4.2 Visualization

Network researchers can find their data represented by numerous talented information visualization specialists. This task can become very technical very quickly, as people seek to represent an underlying structure merely by calculating parts of it in different ways. The Fructerman-Rheingold force directed algorithm gives the classic 'network-ish' aesthetic, leading to insights about subgroups and clusters. By superimposing email networks over a company hierarchy (and measuring the overlap accordingly), Adamic and Adar show how the communication network conforms to the company hierarchy [7]. Representing mainly the dyads and the temporal structure (rather than the network structure) can also be insightful. Viegas and Smith's newsgroup crowds visualization enables one to interpret long-term patterns of reciprocity in chat forums [38]. But perhaps the most striking as of late is Boyack's representation of the 'Web of Science'. By analyzing co-citation patterns in ISI's Citation

Index, he has shown how disciplines as diverse as Organic Chemistry, Sociology and Computer Science are all part of an interconnected, but intelligible web of knowledge [39].

5 Exogenous advances

5.1 Advent of the internet

Perhaps the most obvious, and significant, recent change is the advent of the internet. By allowing us to communicate with digital bits, communication gets encoded merely through its use. That is to say, data sets start creating themselves not because the sociologist asked for them, but because they are part of the maintenance of an internet-oriented communication.

Even something as passive as shopping is grist for the sociological / computational mill. Krebs has been annually producing a series of network diagrams based on the purchasing habits of U.S. liberals and conservatives using only the Amazon API and his inFlow network software.[3]

5.2 Computational power

Both the visualization of networks and the calculation of structural metrics can be a time intensive process. Some important metrics , like betweenness, have only been reduced to $O(n^2)$ time, while others are even $O(n^3)$. Alternative metrics (such as PageRank) help, but they are not a direct substitute given the theoretically meaningful nature of many of the metrics. With advances in computational power, we are beginning to play with our data instead of very rigidly and deductively grinding out a specific set of metrics.

One attempt to leverage this computational power is the ongoing NetWorkBench Cyberinfrastructures project at the University of Indiana[4]. This project is halfway between an algorithm repository and a web services framework for piping large data sets to a central supercomputer for processing.

5.3 Cultural changes

The world at large is becoming more responsive to social network analysis. There are numerous reasons for this. They include the presentation of clever and visually appealing maps [39], the advent of social software (which explicitly requires an individual to demarcate and maintain their network), and the inclusion of network ideas in common vernacular (”six degrees of separation”). As is the case with most sciences, there is still quite a disjuncture between scientific knowledge and lay understanding, but in this field people often ’get it’, and network literacy is, I would surmise, increasing.

One interesting halfway point between rigorous scientific network knowledge and lay understanding is the new phenomenon of data mash-ups. Network data can be piped and displayed using Yahoo Pipes, IBM’s Many Eyes and a host of small java applications. Insights from these simple interfaces may not be the most profound, but they stimulate discussion, and perhaps more importantly raise general network literacy. It is also the case that the interfaces for these tools represent significant improvements over scripting and they may pave the way to more interactive live data analysis in the future.

[3] http://www.orgnet.com/divided.html
[4] http://nwb.slis.indiana.edu/

6 A cohesive programme of network science

6.1 Interdisciplinary collaboration

It is not too much to suggest that there is a emerging cohesive programme of network science, which has many foundations in sociology, but is by no means limited to it. There is presently no professional organization, but NetSci, the International Conference in Network Science is emerging as an interdisciplinary complement to the established social science-oriented International Sunbelt Social Networks Conference. Within this paradigm, the social scientists will most likely use their particular skill sets to frame questions, develop valid research designs and interpret figures reflexively. However, it is unlikely that many will graduate with the technical prowess necessary to implement a rich programme of network science. To complete this programme, we need to employ the aid of others with knowledge of unsupervised learning techniques, relational databases and scripting languages. Dealing with this data is becoming easier through the use of APIs. Most online social forums now have at least a basic mechanism for querying data. This includes sites like Yahoo, Google, Digg and Facebook. The big challenge for sociologists now is to bridge the gap between these lists of data and the usual rectangular data sets necessary for both network analysis and standard regression modeling.

6.2 Data issues within this programme

Accessing and analyzing quality data is an essential but often overlooked condition of possibility for the sorts of analysis described above. Presently, there are few sources for best practices regarding online and social network data. As such, there are still numerous open problems in the area of data quality and retrieval. Below are a list of particular issues that I suggest will become increasingly relevant.

Thresholding: How strong does a tie have to be for the relationship to be meaningful? Thresholding is the process of limiting ties between nodes to those that fulfill a specific threshold of activity. In an email network, one might consider a threshold of 6 messages between individuals in two months as sufficient to assume there is a 'strong tie'. While all authors agree that there is a need to somehow filter out irrelevant mail and spam from the analysis of network data, the specific scope conditions vary from project to project. By using a reciprocal threshold (i.e. a minimum of one message between two correspondents) one can ensure that there is at least some communication - but beyond that is a methodological "no man's land". The same can be said for links on web pages, replies in bulletin boards, calls on a cell phone, etc... Of course, one can keep all ties in the data set no matter how trivial, but then many models of diffusion, influence and community structure might not give answers that are particularly meaningful.

Algorithms for weighted graphs: Thresholding could partly be ameliorated with better algorithms for weighted graphs. There is some work on centrality algorithms for weighted graphs, but the field is still quite unsettled. Interpretations of these algorithms remain at the statistical, and not the substantive, level. One large challenge is the presence of exponential distributions for most measures - there's always a handful who either communicate more frequently, post more often, search more books, etc.

Robust formats for storing large rich data sets: Network analysis has as many data formats as there are programs (if not more). One of the emerging standards is GraphML. However, like all XML files, it contains a significant amount of supplementary text. For massive data sets this additional text scales linearly with an increase in nodes or edges leaving files many times larger than they need to be. Alternative formats such as Pajek are very lightweight but do not do as good a job of ensuring that certain data are associated with particular nodes or edges. Designing a halfway point between the austerity of Pajek and the clarity of GraphML with the ability to conveniently append data, particularly time sensitive data, will be a great improvement.

Better methods for slicing data (particularly temporal slices): Cleaning data is generally an unpleasant experience. For the expert programmer, the SQL queries can be tedious, and for the scripting-challenged, it is down right arduous. Presently, it is done by filtering the interactions which are then processed into networks, not by slicing the networks themselves (that is to say, they are sliced in SQL first, then exported to a network analysis package and then analyzed). A robust object model that maintains a sense of links over time should be able to dynamically slice the network without requiring the researcher to rebuild the network after every slice. Such techniques will enable more efficient sensitivity analyses for thresholds as well as facilitate more exploratory analysis of temporally-oriented network data (such as changes on Wikipedia).

Social network-oriented support in APIs: Support for networks is implicit in numerous APIs. However, this can be leveraged even more successfully (and reduce the number of queries to a site) by anticipating network data. For example, presently if one wishes to capture "Joe's" network on facebook the steps are unnecessary clumsy. First, the program reads all of Joe's ties as a list. To find out who on Joe's list have befriended each other, the program has to then go to (almost) every other list, and compare these lists. By providing an option to query for mutual ties, one can reduce this querying from all of the friend lists of all of Joe's friends to a single list of user-user pairs. This puts an additional processing burden on the server, but it is a simple query for the server, rather than a series of steps for the user (and it reduces data and bandwidth).

People and relations as first class objects: Some frameworks will allow people to be considered first class objects. This allows individuals to be sorted, indexed and have numerous attributes, all of which are easily accessible through the program. However, a significantly harder technical challenge is to design a framework or language that will implement relations between individuals as first-class objects. Obviously, the dependencies between relations and people will make this challenging. But the end result will facilitate easier and perhaps faster querying of relations as well as enable more straightforward code and perhaps even simpler algorithms. It would certainly make network support in APIs dramatically easier to implement.

7 Conclusion

7.1 Concluding thoughts

The field of network analysis has been changing at a blistering rate. There is an influx of talented researchers from a host of disciplines. Network analysis is being done by MacArthur fellows and at the Santa Fe institute. It is featured in the Museum of Modern Art and on numerous blogs. It is an essential part of epidemiological modeling and our notions of social cohesion. Underlying all of this progress is an interest in a deceptively simple type of data that records and tracks links between entities. It has come a long way in the last half a century. With new data sources like the World Wide Web and new tools to examine this data more efficiently, it is likely that we will be busy for the next fifty years at least.

References

[1] J. Gray and A. Szalay, "Where the rubber meets the sky: Bridging the gap betweed databases and science," *Bulletin of the Technical Committee on Data Engineering*, vol. 27, no. 4, pp. 3–11, December 2004.

[2] H. R. Bernard, P. D. Killworth, and L. Sailer, "Informant accuracy in social network data iv: A comparison of clique-level structure in behavioral and cognitive network data," *Social Networks*, vol. 2, no. 3, pp. 191–218, 1979.

[3] j. adams and J. Moody, "To tell the truth: Measuring concordance in multiply reported network data," *Social Networks*, vol. 29, no. 1, pp. 44–58, January 2007.

[4] L. C. Freeman, A. K. Romney, and S. C. Freeman, "Cognitive structure and informant accuracy," *American Anthropologist*, vol. 89, no. 2, pp. 310–325, 1987.

[5] E. Costenbader and T. W. Valente, "The stability of centrality measures when networks are sampled," *Social Networks*, vol. 25, no. 4, pp. 283–307, 2003.

[6] G. Kossinets, "Effects of missing data in social networks," *Social Networks*, vol. 28, no. 3, pp. 247–268, July 2006.

[7] L. Adamic and E. Adar, "How to search a social network," *Social Networks*, vol. 27, no. 3, pp. 187–203, 2005.

[8] L. Adamic and N. Glance, "The political blogosphere and the 2004 u.s. election: Divided they blog," *Working Paper*, 2005.

[9] G. Kossinets and J. Watts, Duncan, “Empirical analysis of an evolving social network,” *Science*, vol. 311, no. 5757, pp. 88–90, 2006.
[10] E. Mayo, *The Human Problems of an Industrial Civilization*. New York, NY: MacMillan, 1933.
[11] d. boyd, “Friends, friendsters and top 8: Writing community into being on social network sites,” *First Monday*, vol. 11, no. 12, 2006.
[12] S. Milgram, “The small-world problem,” *Psychology Today*, vol. 1, no. 1, pp. 60–67, 1969.
[13] M. Granovetter, “The strength of weak ties,” *American Journal of Sociology*, vol. 78, pp. 1360–1380, 1973.
[14] C. Fischer, *To Dwell Among Friends*. Chicago: University of Chicago Press, 1982.
[15] B. Wellman, “The community question: The intimate networks of east yorkers,” *American Journal of Sociology*, vol. 84, no. 5, pp. 1201–1233, 1979.
[16] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.
[17] P. Bonacich, “Power and centrality: A family of measures,” *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.
[18] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Working Paper*, 1999.
[19] M. S. Mizruchi, *The Corporate Board Network*. Thousand Oaks, CA: Sage, 1982.
[20] J. M. McPherson, L. Smith-Lovin, and M. Brashears, “Changes in core discussion networks over two decades,” *American Sociological Review*, vol. 71, no. 3, pp. 353–375, 2006.
[21] E. Rogers, *Diffusion of Innovations, Fourth Edition*. New York: Free Press, 1995.
[22] I. Wallerstein, *The modern world system: Capitalist agriculture and the origins of the european world economy in the sixteenth century*. New York, NY: Academic Press, August 1997.
[23] D. Krackhardt, “Cognitive social structures,” *Social Networks*, vol. 9, no. 2, pp. 109–134, 1987.
[24] K. Faust and S. Wasserman, “Blockmodels: Interpretation and evaluation,” *Social Networks*, vol. 14, no. 1-2, pp. 5–61, 1992.
[25] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
[26] M. E. J. Newman, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8577–8583, 2006.
[27] 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*, vol. 81, no. 4, pp. 730–780, 1976.
[28] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the web,” *Computer Networks*, vol. 33, no. 2, pp. 309–320, 2000.
[29] P. Doreian, V. Batageli, and A. Ferligoj, *Generalized Blockmodeling*, M. Granovetter, Ed. Cambridge, UK: Cambridge University Press, 2005.
[30] S. Wasserman and K. Faust, *Social Network Analysis*. Cambridge, UK: Cambridge University Press, 1994.
[31] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: Simple building blocks of complex networks,” *Science*, vol. 298, pp. 824–827, 2002.
[32] S. Wasserman and P. E. Pattison, “Logit models and logistic regressions for social networks: I. an introduction to markov grahps and p*,” *Psychometrika*, vol. 61, pp. 401–425, 1996.
[33] P. Holland and S. Leinhardt, “An exponential family of probability distributions for directed graphs,” *Journal of the American Statistical Association*, vol. 76, pp. 33–65, 1981.
[34] S. Wasserman and G. Robins, “An introduction to random graphs, dependence graphs and p*,” in *Models and Methods in Social Network Analysis*, P. J. Carrington, J. Scott, and S. Wasserman, Eds. Cambridge, UK: Cambridge University Press, 2005.
[35] D. Watts, *Six Degrees: The Science of a Connected Age*. New York: W. W. Norton, 2002.
[36] A.-L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, 1999.
[37] A.-L. Barabasi, *Linked*. New York: The Penguin Group, 2003.
[38] *Newsgroup crowds and Author lines: Visualizing the Activity of Individuals in Conversational Cyberspaces*, 2004.
[39] K. Boyack and D. Klavans, “Scientific method: Relationships among scientific paradigms,” *Seed Magazine*, vol. 9, pp. 36–37, March 2007.

Managing Uncertainty in Social Networks

Eytan Adar and Christopher Ré
Department of Computer Science and Engineering
University of Washington
{eadar,chrisre}@cs.washington.edu

Abstract

Social network analysis (SNA) has become a mature scientific field over the last 50 years and is now an area with massive commercial appeal and renewed research interest. In this paper, we argue that new methods for collecting social nework strucuture, and the shift in scale of these networks, introduces a greater degree of imprecision that requires rethinking on how SNA techniques can be applied. We discuss a new area in data management, probabilistic databases, whose main research goal is to provide tools to manage and manipulate imprecise or uncertain data. We outline the application building blocks necessary to build a large scale social networking application and the extent to which current research in probabilisitc databases addresses these challenges.

1 Introduction

Though the field of Social Network Analysis (SNA) has developed over the past 50 or more years [21, WF94], it is with the recent emergence of large-scale social networking studies and applications that techniques from this area have received a great deal of public attention. Because the data encapsulated by these networks provides the owners of a system with a mineable resource for marketing, health, communication, and other applications, commercial developers have rushed to construct *social network applications*. Such systems generally enable individuals to connect with old friends and colleagues and form bridges to new individuals in areas ranging from business (e.g. Visible Path [45] and Linked In [30]) to socialization (e.g. Facebook [19] and MySpace [42]) and to entertainment (e.g. iLike [11]). However, translating the research techniques of SNA to large scale applications is a daunting task. With large scale comes imprecision as applications depend on a new set of measurement instruments to collect their data and developers can no longer be completely confident that data about individuals, or the connections between them, is accurate. For example, data collected through automated sensors [9], anonymized communication data (e.g. e-mail headers [1]), and self-reporting/logging on Internet-scale networks [12, 23] as a proxy for real relationships and interactions causes some uncertainty. Furthermore, approximation algorithms [58] intended to calculate network properties (e.g. various centrality measures) on these increasingly large networks creates additional uncertainty. Traditionally, managing large scale datasets has been the domain of data management research and technologies which have almost always assumed that data is precise. In this paper we argue that the transition from research projects to commercial applications creates a need for tools that are able to support SNA techniques and that a critical component is the ability to

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

manage large scale imprecision. Specifically, we make the argument that SNA data can be modeled, managed, and mined effectively by emerging Probabilistic Databases (PDBs). Our discussion does not offer any new implementations or algorithms for PDBs but demonstrates if, when, and how PDBs can be leveraged in the context of SNA and related applications.

The starting point of SNA techniques is a graphical representation in which nodes—called *actors* in the SNA field—represent individuals or groups. An edge (potentially labeled) in this graphical view represents the relationship between actors and generally indicates the possibility of information flow between them. In the early history of SNA, this graph data was collected by survey, interview, and other observational techniques [21, 29, WF94, 57]. While the results were potentially tainted by biased observations, missed observations, and misreporting, the intimate involvement of the researcher (frequently, over extended periods) provided some confidence that the data is precise. As those studying and utilizing social networks have moved to enormous scales, they have frequently sacrificed some accuracy as careful methodologies have become increasingly difficult or impossible. Furthermore, in wild and uncontrolled environments such as the Internet, biases can develop due to application design (e.g. default friends on MySpace) and malicious individuals (e.g. spammers building network connections in some automated way). The result of this "noise" is the introduction of tremendous levels of uncertainty in the data which are ill-supported by current large scale data management systems.

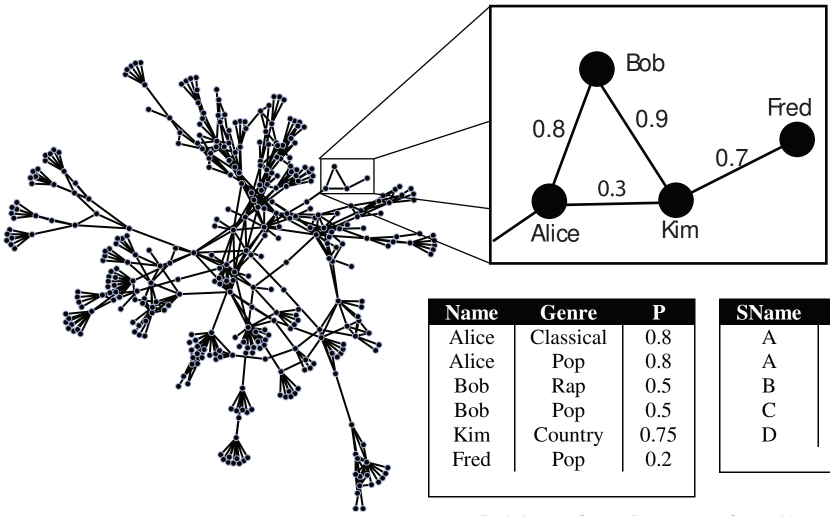
Probabilistic relational databases—the potential answer to these issues—have attracted renewed interest in the data management community [8, 14, 17, 22, 47, 52, 59]. A probabilistic database works much in the same was as a standard relational database, in which tuples (i.e. rows of data) can be stored, searched and aggregated in various ways using SQL. The defining characteristic of a probabilistic database is that to any tuple t, a probability is associated that indicates the probability t is in the database. While a standard relational database is intended to support a precise data model (e.g. Bob lives at "121 South Street"), a probabilistic database models uncertainty (e.g. there is 80% chance Bob lives at "121 South Street" and a 20% chance he lives at "50 West Street"). The motivating goal of this area is to provide application developers with the tools they need to *manage* imprecision while providing industrial scale performance. In this paper, we select a very simple data model called *tuple independence*, which is supported by all models in the recent literature. We refer the interested reader to work on more sophisticated models that are capable of representing any distribution (e.g. [47, 51, 52]) and research on models dealing with continuous data (e.g. sensor networks [8, 17]).

2 A Motivating Example

To understand the application of probabilistic databases (PDBs) in the context of social network research, we concentrate our efforts on a fictitious diffusion model[1] for music recommendations. Diffusion models are interesting in that they capture a range of application areas including epidemic models (e.g. [36, 40, 43]), innovation diffusion (e.g. [5, 10, 50, 54, 55]), and rumor and gossip propagation (e.g. [33]). Diffusion models are additionally relevant to both pure scientific discoveries about basic behavioral processes (e.g. [10, 25, 41]) and applied endeavors such as expert-finding networks [2], recommender systems [32], and public health community-building [5]. We later return to some of these applications by generalizing our example to a broader class of diffusion models.

Our system, graphically represented in Fig. 1, has two types of data about its users: standard actor/node information (e.g. name, age, residing city, etc.) (Fig. 1(b)) and preference data (e.g. music preferences) in terms of genre (Fig. 1(c)). Because we have determined the preference through a sampling methodology (e.g. by asking individuals to indicate their like or dislike of a set of songs), we are uncertain about its true value. This is modeled by assigning a probability to a tuple (e.g. (Kim,Country) is in Prefs with probability 0.75). To simplify our model, we assume that tuples that do not exist have a probability 0 (e.g. Alice does not like rap

[1]A full survey of this field is well beyond the scope of this paper. These citations represent interesting exemplars in this space. Some are early, influential publications, others represent more modern examples.



Name	Age	City
Alice	25	Seattle
Fred	21	Reno
Bob	30	Seattle
Kim	27	LA

Users(Name,Age,City)
Figure 1b

Name	Genre	P
Alice	Classical	0.8
Alice	Pop	0.8
Bob	Rap	0.5
Bob	Pop	0.5
Kim	Country	0.75
Fred	Pop	0.2

Prefs(Name, Genre, P)

SName	Genre	P
A	Country	0.1
A	Pop	0.5
B	Rap	0.5
C	Classical	0.8
D	Country	0.6

Songs(Name, Genre, Probability)

Name1	Name2	P
Alice	Bob	0.8
Alice	Kim	0.3
Bob	Kim	0.9
Fred	Kim	0.7

MusicInfuence(Name1,Name2,P)

Figure 1a Figure 1c Figure 1d Figure 1e

Figure 1: Sample Data for Social Network Integration

```
SELECT U.name
FROM Users U, Prefs P, Songs S
WHERE U.name = P.name
  AND P.Genre = S.genre
  AND S.name = 'A'
```

(a)

```
SELECT U.Name
FROM Users U , Prefs P, Song S,
  MusicInfluence M, Recommends R
WHERE U.name = P.name AND P.Genre = S.genre
  AND M.name2 = U.name AND M.name1 = R.from
  AND R.To = U.name AND R.song = S.sname
  AND S.sname = 'D'
```

(b)

Figure 2: Sample Queries

music)[2]. Additionally, we have a table of songs (Fig. 1(d)). It is often not clear-cut to which genre a songs belongs, which we model by assigning a probability that a song belongs to a given genre. Given this data we can now ask questions of the form: for song A in Songs, "what is the probability that a user would like A?" This is expressed in SQL in Fig. 2(a). For example, the user Kim would have probability $0.75 * 0.1 = 0.075$ since we have assumed all tuples are independent (note that result probabilities are always returned alongside result tuple without the user needing to make an explicit request for those).

Consider an additional table, MusicInfluence, shown in Fig. 1(e), that describes a piece of our social network. To construct this table, we have assumed that users have explicitly defined their network (i.e. Alice has indicated that she is friends with Bob), and through some experience, we have assigned a probability on each edge indicating the likelihood that a musical recommendation from the first user will be picked up by the second (e.g. of the previous n recommendations, k were accepted. In our example, Alice influences Bob, with probability 0.8). We can use this table to write more interesting queries. Assuming that both Alice and Bob have recommended song D to Kim (Fig. 3a), we can ask, "what is the probability that Kim will be affected by these recommendations?" Intuitively, the SQL query Fig. 2(b)) is satisfied for Kim when, song D is country (0.6), Kim likes country (0.75) and either transmission from Alice or Bob is successful (0.8 and 0.9, resp.). Thus, Kim

[2] [4] considers a semantic which is able to account for missing data as an "unknown" or "wildcard."

From	To	Song
Bob	Kim	D
Alice	Kim	D
...		

Recommends(From,To,Song)

(a) Recommends(from,to,song)

Name1	Name2	P
Alice	Bob	.8
Alice	Kim	.3
Bob	Fred	.1
...		

MusicSim(Name1,Name2,P)

(b) MusicSim(name1,name2;P)

Figure 3: Sample Data for Social Network Integration

is returned with a probability value of $0.6 * 0.75 * (1 - (1 - 0.8) * (1 - 0.9)) = 0.441$.

For the sake of exposition, we have simplified our example considerably; there are much more powerful types of models are possible: e.g. we could extended table to have the probability be given by type of music (e.g. (name1, name2, genre, probability)) or have more complicated correlations between the tuples (e.g. using *factors* [52] or BID tables [47]).

2.1 Application Building Blocks

In this section, we highlight some of the fundamental requirements of building a large scale SNA application and discuss the extent to which probabilistic databases can help address these requirements.

Data Analysis One of the most (if not the most) important business aspects of a social networking site is understanding the network. Put simply: you can not monetize what you do not understand. For example, in our scenario we may be interested in distributing free concert tickets for a new artist to a small subset of users. Since there is a cost to providing these tickets, we would like to find a small group of consumers who have a large amount of influence, i.e. a set of *trend-setters* or *influencers*. In the SNA world, this is a similar to a query for nodes with a high degree-centrality measure (e.g. the number of outgoing edges is high). Since the data are uncertain, one natural semantic is to rank users by their expected number of outgoing outgoing edges. These are essentially probabilistic *decision support* or OLAP style queries [7, 31]. Alternatively, we may only want to send the tickets to high value users, e.g. those with a high probability of having more than k edges, which has been consider in [48]. Although a large class of these queries can be handled efficiently by probabilistic databases, adapting more sophisticated SNA algorithms is an interesting direction for future research.

Scalability Large scale social network applications have very large datasets which need to be manipulated with good performance. Continuing our previous example, our webpage would need to issue queries such as "which users I am most similar to?" or "which users am I most similar to who live in Seattle?". This is daunting because the queries can combine several sources of probabilistic information. Sometimes, it is possible to correctly compute probabilistic database queries directly in SQL using a new technique called *safe plans* [14, 15, 46]. Intuitively, safe plans tell us when a probabilistic query can be computed by simply multiplying (and summing) probabilities. However, safe are not always possible, in which case a query may require approximation algorithms (e.g. a restricted kind of Monte Carlo simulation), which are slower but still tenable at large scales [47]. Often we are only interested in computing the top k answers, which can allow substantial improvements [47, 53]. Further, new research suggests that we may be able to materialize a probabilistic view which allows complex probabilistic datasets to scale even in to the tens and hundreds of gigabytes [49]. While probabilistic database research is still in its infancy, there are already techniques to scale up to huge datasets.

Physical and Semantic Independence In a large social network, as in any application with large numbers of users, tuning the backend is critical and requires constant tinkering. This effort is mitigated by a property that probabilistic databases on relational models inherit from relational databases, *physical independence*. PDBs achieve physical independence, because all interaction takes place through a query language that does not reference the physical layout on disk. Hence, data can be partitioned and indexed independently of how they are used by application code. Also important to sites that offer recommendations is the ability to compute, and propagate, qualitatively good recommendations. Thus, an approach is untenable if changing the code that computes the influence probability requires changing the code that displays the top ten most similar users. PDBs mitigates this problem because they achieve *semantic independence*. In particular, tuples have a clear probabilistic semantic independently of how they are computed. A tangible benefit is that we can decouple the computation of probabilities from their use in application code.

Maintainability A major problem in any large scale enterprise is maintaining, updating and debugging the data and applications built on it. As a concrete example, if the data, on which recommendations are based, changes (e.g. a user submits that they like new genre), the values in the relation should change as well. Also, if the end result of the computation breaks, how do we know how to fix it? There is very promising work in this direction based on *lineage* [51] in uncertain databases, which helps an administrator understand why or how a probability value is computed. We feel that the large body of work in the AI community on explaining a probabilistic proposition is a good starting place (e.g. [6, 35, 38], but one key remaining challenge is scaling these techniques to large datasets.

Integration Merging social networks is interesting from a research perspective as well as a business perspective [34]. For example, consider merging the network described above and an independent friendship network (e.g. Facebook). Intuitively, by leveraging more information the merging of two networks should provide higher answer quality and also allows us to ask queries not answerable by either network alone. For example, suppose we want to sell concert tickets for an intimate venue that only sells tickets in blocks of four (e.g. for tables). We would like to know, which users have three friends with similar tastes in music and live in the same area. To do this, we need to know both a persons friends and their taste in music, information not available in either networks by themselves. There are many difficult problems in integration, e.g. *entity resolution* or *reference reconciliation* [18, 20, 27, 61]. However, we believe a probabilistic databases provides a solid framework to model the uncertainty of inherent in the integration process.

Handling Missing Data While we may have an explicitly defined network it is always possible that we are missing certain important edges. This may be due to misreporting or flawed instrumentation but the outcome is an incomplete network. The idea that edges can be inferred has been studied extensively (e.g. by [24]). For our example we may use a simple algorithm that calculates the pairwise similarity between individuals based on the musical preference (and potentially their neighborhood). We model the output of the matching procedure using a probabilistic relation, which we call MusicSim. A snippet of data is shown in Fig. 3(b). A tuple in MusicSim means that name1 and name2 are similar, with probability given by the P attribute. For example, we might find that Alice and Bob have similar music tastes with probability 0.8, but Alice and Kim have similar music tastes with only probability 0.3. This is a powerful idea that is simplified greatly by the use of a probabilistic database.

2.2 Beyond Music

Though we have concentrated on a specific type of diffusion network above, there are clearly many application areas beyond music recommendation. An epidemic model, for example, may take into account susceptibility to a certain disease based on individual features, transmission probabilities assuming repeated contacts, probabilities of immunization and other complex dynamics ([44]). A corporate network may take into account hierarchical and managerial influence on adoption of innovations. Clearly, correctly generating such models is a difficult and time consuming task, but managing and querying this type of imprecise information—especially in large scales—may be aided by the use of a probabilistic database.

3 Additional Application Areas

There are many additional areas in which social network analysis and applications are starting to be utilized in which the data is inherently imprecise. We select two of these areas that we think present particular important and interesting and where probabilistic databases have already received some attention.

Privacy and Anonymization As the use of social network information becomes more prevalent, it is important to recognize the privacy concerns of individuals. To understand the implications of social networks for privacy rights, a number of researchers [3, 26] have begun to explore how social networking data can, or can not, be anonymized using data perturbation techniques. We believe that probabilistic databases can play an interesting role in moving theoretical techniques of privacy-preservation (e.g. [13, 37, 39]) into large scale applications.

Homeland Security A sub-area of SNA that recently received a lot of attention is the analysis of terrorist networks. Here, SNA is focused on identifying "critical" individuals in the network. A report to congress by DARPA [16] about the now defunct Genisys program, highlighted the inadequacy of standard relational databases for the task and the need for "probabilistic database representing and dealing with uncertainty". Interestingly, the program was part of the TIA project that was defunded due to privacy concerns. However, according to media reports essentially the same program is still funded, under the name Topsail [28, 60].

4 Conclusion

In this paper we have argued that probabilistic databases are a useful paradigm for those who want to build social networking applications. The inherent imprecision and uncertainty of large-scale social network analysis, both in collection and analysis, does not need to add tremendous complexity to researchers and application designers. Even in their nascent state, probabilistic databases have much to offer social networking analysis and applications by handling the models, scaling, maintenance and analysis needs. Furthermore, we believe that social networks are an important motivating application for probabilistic database research. The growth of research and economic interest in social networking applications has generated a tremendous set of potential consumers of probabilistic databases. We have briefly discussed a number of interesting open research and technical problems to enable and support a wider range of social network applications. A mutually beneficial relationship between these two communities, especially during the rapid growth in both domains, will likely lead to many novel algorithms, techniques, and systems beyond anything we have imagined in this paper.

5 Acknowledgements

The authors would like to thank Bernie Hogan, Lada Adamic, Dan Weld, and Mike Cafarella for their comments and discussions. Eytan Adar is funded by an ARCS and NSF Graduate Fellowship.

References

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

[2] E. Adar, R. Lukose, C. Sengupta, J. Tyler, and N. Good. Shock: A privacy-preserving knowledge network. *Information Systems Frontiers*, 5(1), 2003.

[3] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou R3579X? anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of WWW 2007*, 2007.

[4] D. Barbará, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[5] D. M. Berwick. Disseminating innovations in health care. *Journal of the American Medical Association*, 289(15), 2003.

[6] A. Borgida, E. Franconi, and I. Horrocks. Explaining ALC subsumption. In Werner Horn, editor, *ECAI*, pages 209–213. IOS Press, 2000.

[7] Douglas Burdick, Prasad M. Deshpande, T. S. Jayram, Raghu Ramakrishnan, and Shivakumar Vaithyanathan. OLAP over uncertain and imprecise data. *VLDB J.*, 16(1):123–144, 2007.

[8] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of ACM SIGMOD Conference*, 2003.

[9] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester. Towards activity databases: Using sensors and statistical models to summarize people's lives. *IEEE Data Eng. Bull.*, 29(1):49–58, 2006.

[10] J. Coleman, E. Katz, and H. Menzel. The diffusion of an innovation among physicians. *Sociometry*, 20(4), December 1957.

[11] Garage Band Corp. `www.ilike.com`.

[12] d. m. boyd. Friendster and publicly articulated social networking. In *Proceedings of CHI 2004*, pages 1279–1282, 2004.

[13] N. Dalvi, G. Miklau, and D. Suciu. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*, 2005.

[14] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, Toronto, Canada, 2004.

[15] N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In *PODS*, 2007.

[16] DARPA. Report to congress regarding the terrorism information awareness program,
`http://www.epic.org/privacy/profiling/tia/may03_report.pdf`, 2003.

[17] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks, 2004.

[18] X. Dong, A. Y. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In Fatma Özcan, editor, *SIGMOD Conference*, pages 85–96. ACM, 2005.

[19] Facebook. `www.facebook.com`.

[20] I. Felligi and A. Sunter. A theory for record linkage. *Journal of the American Statistical Society*, 64:1183–1210, 1969.

[21] L. C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.

[22] A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. In *ICDT*, pages 337–351, 2005.

[23] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3, 1997.

[24] L. Getoor and C. P. Diehl. Link mining: A survey. *SIGKDD Explorations*, 7(2), 2005.

[25] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6), May 1978.

[26] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical Report 07-19, University of Massachusetts Amherst, CS Department, March 2007.

[27] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In Michael J. Carey and Donovan A. Schneider, editors, *SIGMOD Conference*, pages 127–138. ACM Press, 1995.

[28] M. Hirsh. Wanted: Competent big brothers. *MSNBC*,
`http://www.msnbc.msn.com/id/11238800/site/newsweek/`, Feb 2006.

[29] B. Hogan, J. A. Carrasco, and B. Wellman. Visualizing Personal Networks: Working with Participant-aided Sociograms. *Field Methods*, 19(2):116–144, 2007.

[30] Linked In. `www.linkedin.com`.

[31] T.S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *SODA*, 2007.

[32] H. Kautz, B. Selman, and M. Shah. Referral web: combining social neworks and collaborative filtering. *Communications of the ACM*, 40(3), March 1997.

[33] A. C. Kerckhoff and K. W. Back. Sociometric patterns in hysterical contagion. *Sociometry*, 28(1):2–15, March 1965.

[34] D. Kirkpatrick. Facebook's plans to hookup the world. *Fortune Magazine, online edition.* `http://money.cnn.com/2007/05/24/technology/facebook.fortune/index.htm`, 2005.

[35] N. Kushmerick. Regression testing for wrapper maintenance. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, Florida, July 1999. Menlo Park, CA: AAAI Press.

[36] A. L. Llyod and R. M. May. How viruses spread among computes and people. *Science*, 292(5520), May 2001.

[37] A. Machanavajjhala and J. Gehrke. On the efficiency of checking perfect privacy. In Stijn Vansummeren, editor, *PODS*, pages 163–172. ACM, 2006.

[38] D. McGuinness and A. Borgida. Explaining subsumption in description logics. In Chris Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 816–821, San Francisco, 1995. Morgan Kaufmann.

[39] G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In *SIGMOD*, 2004.

[40] M. Morris. *Network Epidemiology: A Handbook for Survey Design and Data Collection*. Oxford University Press, 2004.

[41] M. Morris and M. Kretzschmar. Concurrent partnerships and the spread of HIV. *AIDS*, 11(5):641–648, 1997.

[42] MySpace. `www.myspace.com`.

[43] M. E. J. Newman. Spread of epidemic disease on networks. *Phys. Rev. E*, 66(1):016128, Jul 2002.

[44] M. A. Nowak and R. May. *Virus dynamics: Mathematical principles of immunology and virology*. Oxford University Press, 2001.

[45] Visible Path. `www.visiblepath.com`.

[46] C. Ré, N. Dalvi, and D. Suciu. Query evaluation on probabilistic databases. *IEEE Data Engineering Bulletin*, 29(1):25–31, 2006.

[47] C. Ré, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *Proceedings of ICDE*, 2007.

[48] C. Ré and D. Suciu. Efficient evaluation of `HAVING` queries on probabilistic databases (full version). Technical report, University of Washington, Seattle, Washington, June 2007.

[49] C. Ré and D. Suciu. Materialized views in probabilsitic databases for information exchange and query optimization. In *VLDB '07 (to appear)*, 2007.

[50] B. Ryan and N. C. Gross. The diffusion of hybrid seed corn in two iowa communities. *Rural Sociology*, 8(1), 1943.

[51] A. D. Sarma, O. Benjelloun, A. Y. Halevy, and J. Widom. Working models for uncertain data. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *ICDE*, page 7. IEEE Computer Society, 2006.

[52] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *Proceedings of ICDE*, 2007.

[53] M. Soliman, I.F. Ilyas, and K. Chen-Chaun Chang. Top-k query processing in uncertain databases. In *Proceedings of ICDE*, 2007.

[54] T. W. Valente. *Network Models of the Diffusion of Innovations*. Hampton Press, 1995.

[55] T. W. Valente and R. L. Davis. Accelerating the diffusion of innovations using opinion leaders. *Annals of the American Academy of Political and Social Science*, 566, November 1999.

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

[57] B. Wellman. Challenges in Collecting Personal Network Data: The Nature of Personal Network Analysis. *Field Methods*, 19(2):111–115, 2007.

[58] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of KDD 2003*, pages 266–275, 2003.

[59] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR*, pages 262–276, 2005.

[60] M. Williams. The total information awareness project lives on. *Technology Review*, April 2006.

[61] W. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

Challenges in Searching Online Communities

Sihem Amer Yahia
Yahoo! Research

Michael Benedikt
Oxford University

Philip Bohannon
Yahoo! Research

Abstract

An ever-growing number of users participate in online communities such as Flickr, del.icio.us , and YouTube , making friends and sharing content. Users come to these sites to find out about general trends – the most popular tags, or the most recently tagged item – as well as for more specific information, such as the recent posts of one of their friends. While these activities correspond to different user needs, they all can be seen as the filtering of resources in communities by various search criteria. We provide a survey of these search tasks and discuss the challenges in their efficient and effective evaluation.

1 Introduction

Online communities such as LinkedIn, Friendster, and Orkut attract millions of users who build networks of their contacts and utilize them for social and professional purposes. Recently, online *content* sites such as Flickr, del.icio.us, and YouTube have begun to draw large numbers of users who contribute content – photos, urls, text and videos. They also annotate the content: tagging it with appropriate keywords, rating it, and commenting on it. A key feature distinguishing these sites from previous content-management sites is the effective integration of the user's social network into the experience of exploring and tagging content. Similarly, some of the most popular online communities such as MySpace and Facebook encourage content-sharing as well as contact-making. As a result, a variety of popular online communities have a rich body of data comprised of user-contributed content, user relationships, and user ratings. We call a Web site supporting such a community a *social content* site.

The functionality of social content sites is based on data generated by users. Users spend their time browsing content and using keyword search to look for interesting content, people who share their tastes, and content posted by like-minded people. Hotlists of new/popular content, keywords, or recommendations may also be offered to users. In all these cases, the user is presented with lists of ranked content. It is critical that the ranking of results has the ability to leverage all the user-generated content and social connection information.

However, ranking of search results over the data on a social content site is far from trivial. First, search needs to take into account *social activity*. For example, if the user types "sunset" on a social photo-sharing site, a social-aware search should account in the ranking of results the rating of the photo by users, the tags of the photo, and potentially for each tag the status or reputation of the tagger – is this person's own photos tagged or endorsed? Second, search results need to be *personalized* based on *the user's context*. The user's contributions include many implicit and explicit indicators of interest – tagging, rating, and browsing activities; friendships, and the activities of friends. While in traditional Web search one *may* wish to utilize user information to enhance

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

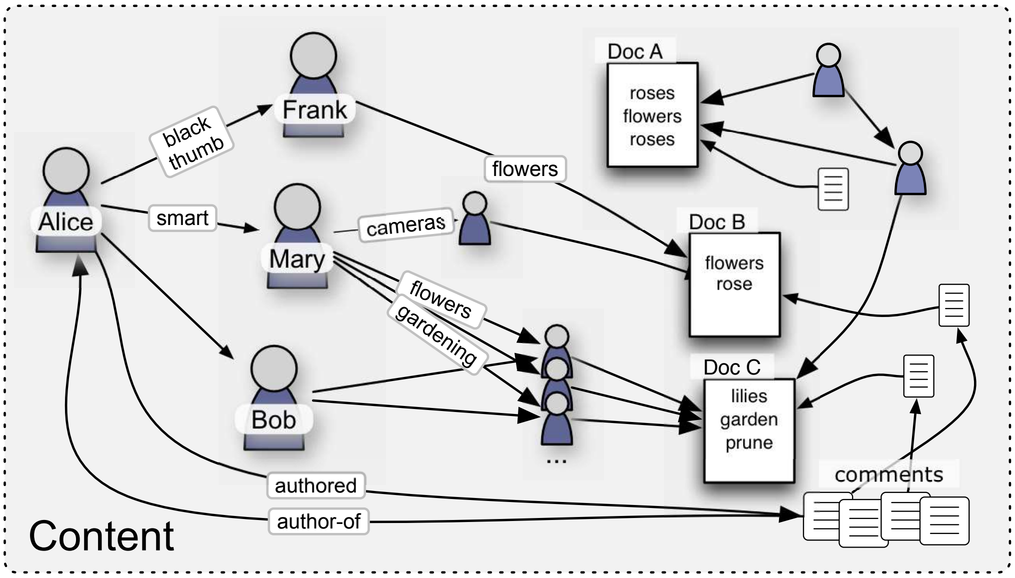


Figure 1: Gardening Social Content Example

search effectiveness, in social content search this information is readily available and essential for meeting the querier's expectation. For example, if the user searches for "birthday party" on the same photo-sharing site, it is reasonable to boost the rank of results from within the querier's social circle. Finally, when displaying hotlists of content or keywords, *recency* is often an important factor, requiring dynamic incorporation of new content in a manner similar to news search (e.g. [New]). Our first contribution is a classification of ranking factors in social search and illustrated with examples drawn from existing social content sites.

Given a particular ranking method, the next critical issue is the *efficiency* with which results can be computed. Obviously, techniques used for Web search are scalable, and have dealt successfully with the astronomic growth of the traditional Web. However, it is far from obvious that social intent, personalization, and recency can be incorporated into search without sacrificing efficiency. We discuss efficient and effective search in Section 3. We conclude and discuss some future challenges in Section 4.

2 Workload and Relevance Factors

In this section, we describe the *relevance factors* that tend to be operative in social content search. We then survey some of the functionality of existing social content Web sites in terms of these factors. In general, we consider three kinds of search targets: *content*, hot keywords, and *people* (usually called *expertise* search [MA00, KSS97]). We will use the term *resource* to refer to any of these search targets.

To illustrate the issues involved in relevance computation, we consider an example fragment of a hypothetical social-content Web site concerned with gardening, displayed in Figure 1. In this example, there are users and two kinds of content, blog posts and comments on those posts. Users can establish links with each other, and may assign a label to their relationship. For example, there is such a link between Alice and Frank. Note that this link is tagged with "black thumb", indicating that Alice has a low opinion of Frank's gardening skills. Users can tag content (dashed arrows), as Frank has tagged Document B with "flowers". Content can also refer to other content via hyperlinks (solid arrows). The right hand side of the figure shows examples of comments referring to documents, with these documents referring to the three named documents, A, B and C.

Whether the user is navigating to a hotlist of resources, browsing another user page or performing a keyword

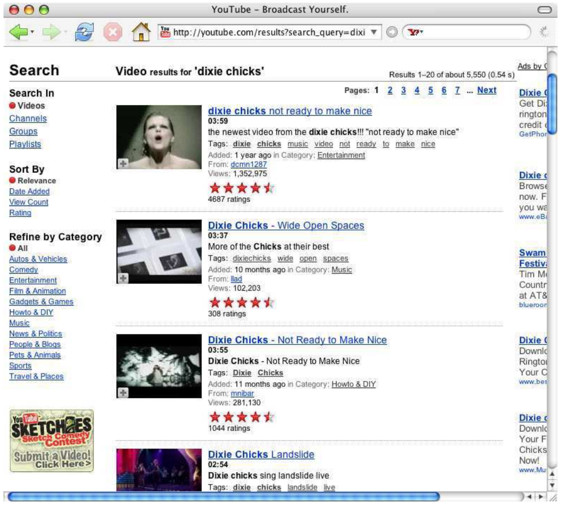


Figure 2: YouTube

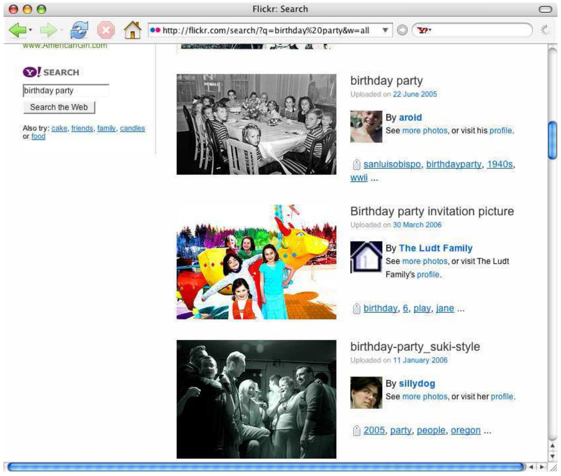


Figure 3: Flickr “Birthday Party” Search

search, the goal is the same: to return a list of resources ranked by score. The score of a resource can be informally defined using any subset of the following factors:

Text Features. When a keyword query is posed, the text associated with any content resource can be scored using standard metrics of relevance such as TF-IDF [SM83]. For multimedia content, there is often a title and description associated with it that can be matched against the keywords. For example, in Figure 1, Document A has content “roses” and “flowers”, with “rose” repeated to show a high TF-IDF score.

Timeliness and Freshness. A resource may be more interesting if it is itself recently added. In the case of content , a simple interpretation of timeliness is as the inverse of the time elapsed since it was posted. One can also measure the timeliness via the popularity of the text associated with the resource – whether or not it is tagged with “hot” keywords.

Incoming Links and Tags. Tags associated with resources are usually a strong indicator of meaning. The anchor text on hyperlinks plays a role analogous to tags. We refer to either kind of link as an *endorsement*. Of course, PageRank-style metrics can be used to measure transitive endorsement - we discuss a variety of different ways such metrics can be computed and used below.

Popularity. A more subtle interpretation of timeliness may consider second-order recency or “buzz” - how much recent tagging and linking activity has targeted a resource? For example, if the references to document A from users on the right had been established in the last hour, A might be considered “hot”. Further, measures of how many times an object is viewed may be incorporated in ranking.

Social Distance. A content resource can be considered “socially close” to the querier if it is tagged by individuals in the querier’s social network. For example, a restaurant tagged by an acquaintance or by a reviewer that the querier has listed as a friend may be more interesting, since the querier may trust (or at least understand) the recommender’s taste. Social distance can be computed by associating a score with each inter-personal link, and using the weight of paths from the querier to some target content. In Figure 1, there is a social connection from Alice, who might issue a query like “flowers”, to document B of only two links through Frank, but the initial edge to Frank may have a low weight due to the tag. The paths from Alice to Document C are longer, but more paths exist, and the initial weights of the edges to Mary and Bob are stronger. Such factors must be balanced when estimating social distance.

Relevance for People. In the case a resource is a person, the documents authored by the person and the

Figure 4: MyWeb Humor Search

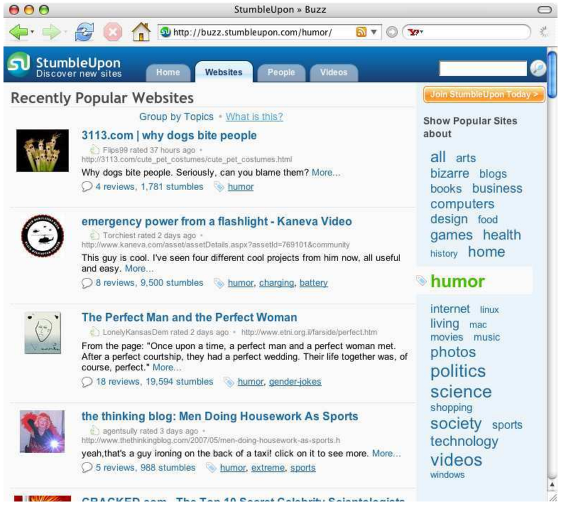


Figure 5: StumbleUpon Web List

outgoing tags can be considered as indicators of the person's *interests*, and thus can play a role analogous to text features of a content resource. For example, the comment authored by Alice may serve as a indicator for her. Inward links can also be important. For example, a person whose blog posts are well-reviewed or which are frequently tagged with words related to gardening might be considered an *expert* in the topic, and new posts by this individual about gardening might thus receive a higher rank.

We now give examples of social search in existing sites and relate them to the relevance factors discussed above.

Examples from Social Sites. One task is certainly keyword search. A straight keyword query can be used to search videos in YouTube – as in Figure 2, which shows the result of the query "dixie chicks". Clearly, ranking should take into account text features such as title and description, along with incoming tags applied to videos weighted by the popularity of the tags. One could envision adding a user's social activity, e.g., to rank videos tagged by friends higher. For blog posts, recency and popularity can be combined with text features [BK07].

Another typical task is browsing content resources – by tag, or by user. Even in browsing, ranking is important, since a given user or tag may have a large quantity of associated contributed content. For example, a user may browse photos related to the tag "birthday party" in Flickr, arriving at the result page shown on Figure 3. It may well be that the user will be more satisfied with photos of recent birthday parties by friends, in which case social distance needs to be accounted for. In some sites, that choice is left to the user - for example MyWeb gives the choice of searching "the Web" or "my contacts" (friendship network). Figure 4, shows the result of searching for "humor" over resources from a user's network. Note that the number of views and saves from the user's network are overlaid with each answer.

Finally, the search may not have keywords at all. That is, content may be *recommended* to the user (see, for example, [HKTR04]). StumbleUpon recommends "hot" resources of the moment (Figure 5), with a fresh list provided each time the page re-loads. The intention of a hotlist may be to show something popular, or to show something that is interesting based on past user clicks. It may even be simply trying to show the querier something in order to test the quality of an unrated content resource.

From these examples, we see clearly that the relevance factors defined in the previous section can be combined in a number of useful ways to power real search functionality in modern social-content sites. In the next section we discuss possible techniques and a number of open challenges in implementing this range of features into effective and efficient search functionality on social content sites.

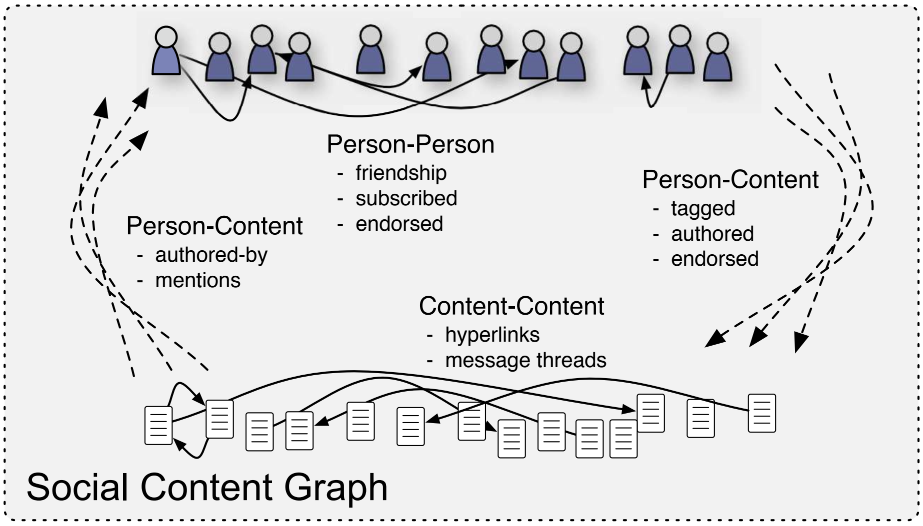


Figure 6: The Social Content Graph

3 Search Efficiency and Effectiveness

In this section, we outline techniques applicable to ranking search results in a social-content graph.

3.1 Integrated Approach

In attempting to capture the interactions between content linking and endorsement from friends, it is natural to treat resource endorsements, friendship links and people endorsements of content uniformly as edges in a "social-content graph". One such integrated approach is to model the relevance requirements of social content sites by parameterizing the behavior of a "random surfer" within a social-content graph, applying variants of PageRank [BP98] or HITS [Kle99] to compute the relevance of person or content nodes to a user's query. We now discuss the elements of this approach.

The Social-Content Graph An example of the directed graph that underlies this approach, called a "social-content graph," is shown in Figure 1, and the general form of such graphs is shown in Figure 6. A social-content graph has two types of nodes, corresponding to people and content (text, photos, videos, etc.). Edges may have associated text (e.g. tags). The semantics of edges in the graph depends on the type of the source and target node. Person-to-Person edges represent endorsement, friendship, or some other social relationship. The text of the edge may come from an explicit tag or the category of the relationship (e.g. "family", "coworker"). The Person-to-Content edges capture tagging, authoring, etc. Text associated with these links is derived naturally from tags. Content-to-Content edges may be hyperlinks, or may represent threading relationships between comments. Text associated with the link may be taken from the anchor text of the hyperlink. Finally, the Content-to-Person edges may show authorship or reference. For example, a search engine might use named-entity recognition (see, for example, [ZPZ04]) to identify references in picture titles to people's names, and establish Content-to-Person links from the picture to the person. Examples of each type of edge can be seen in the example of Figure 1.

Transition Probabilities We now consider how to model a "random surfer"[BP98, Kle99] traversing a social content graph. In this framework, a surfer begins at an arbitrary node in the graph. At each step, the probability of jumping rather than following an outgoing edge is called the *damping factor*. If the user decides to jump, she goes to any node according to its *node weight*. If the surfer follows a edge, then it picks any particular outgoing edge with the probability proportional to an *edge weight*. Modifying the *node weight* based on user preference or on the keywords of the search leads to a *personalized* or *topical* PageRank computation [JW03, Hav02], while modifying the edge and node weights together to reflect the match of content with a query is performed in "intelligent surfing" [RD02]. Note that the *damping factor* controls the *locality* of each "surf".

From Probabilities to Ranking Given a social-content graph, a querier and a (possibly empty) set of keywords, an integrated approach to relevance computation proceeds as follows. First the system assigns node and edge-weights to be used for random surfing to nodes and edges in the social content graph. Second, the stationary probability that a surfer arrives at each node is computed (e.g. a PageRank computation). Third, the k nodes with the highest such probability are returned to the user.

From Relevance Factors to Probabilities We now briefly describe how each of the relevance factors discussed in the previous section might be handled by adjusting the parameters of the computation. First, *text features* can be handled by computing the query relevance of the text associated with a resource to the query terms and setting *node weights* proportional to this relevance. To handle *timeliness and freshness* as well as *popularity*, edges and node weights can be adjusted by their recency [BBVW06]. To handle *incoming links and tags*, the edge-weights can be adjusted to reflect relevance of the query to the tag or other text associated with the edge in the social-content graph (see [BWF$^+$07]). Finally, *social distance* is incorporated naturally in this model by applying a significant fraction of the total node weight to the querier's node and adjusting Person-to-Person edge weights according to the strength of the connection. While node and edge weights can be set individually, coarser parameters will make these easier to manage. For example, it may be helpful to adjust the overall weight of person nodes vs. content nodes for random jumps or of "friends" vs. "family" edges when following a link.

Feasibility and Performance While the integrated approach is conceptually clean and general, there are substantial feasibility and performance issues. As with all variants of PageRank, the stationary probability can be calculated using a fix-point algorithm. But given that the probabilities are only known at query-time, one cannot compute this off-line. One response to this problem is to use random surfing for only a subset of the features, relying on existing Web techniques for the remainder. For example, if one excludes hyperlink structure, one arrives at a graph that is considerably smaller than Web graphs. This *modular* approach is discussed in the next section. Another route is to apply recent work on accelerating dynamic PageRank [FR04, Cha07]. A technique to address the social-distance component is to approximate [CGS04] PageRank values only for nodes in the neighborhood of the querier. However, this approach is complicated by the fact that social graphs obey a so-called "power-law" distribution [WS98, New00, WF94], meaning that individual neighborhoods may be relatively large on average.

3.2 Modular Approach

The integrated approach gives the possibility of accounting for social distance in a very fine-grained way. It can account for the propagation of authority from a socially-close user through resource hyperlinks. But it certainly lacks modularity, since it cannot exploit the components already developed for Web search, particularly in the area of content and page-quality scoring.

A more modular but coarser approach is to consider each factor (or a subset of the factors) in isolation, coming up with separate rankings that are averaged to get one final score. Consider a case where a resource is content, the query-relevance could be done by traditional TF-IDF scoring of the content, and the resource endorsements could be ranked via PageRank or some other link analysis method. This does not eliminate the issue of incorporating social distance, but it does reduce it to two main components. The first is calculating a social endorsement score, which should average the impact of query-matching tags from friends in a user's network; The second issue is the overall combination problem, which should result in a single score formed by combining component scores over the various dimensions e.g., by taking a weighted average. The second problem revolves around the choice of weights, which we discuss in Section 3.4.

The gain in modularity in this approach is counter-balanced by a possible loss in effectiveness, since each factor is now considered in isolation. Consider a page resource that is not itself tagged by many in the user's community, but which is linked to many pages that are tagged by many in the community: such a resource might score low both in link-quality and tagging quality, although it is likely to be quite relevant.

3.3 Computing Social Endorsement

To see the daunting efficiency issues that remain in the calculation of the social endorsement score, consider a simple example where the query is a keyword search and the social endorsement score of a resource is computed as the number of times friends of the querier have tagged the resource with query-matching tags. One must consider two related issues here: what sort of indexing structure is available for the tagging information, and the query evaluation algorithm. The most natural approach is to organize data in inverted lists by tag, in direct analogy with what is done for terms in a standard IR setting. Each entry in the list records a resource identifier and also its list of taggers.

Given a query, the score of a resource could be computed as the number of its taggers who are friends of the querier, or as the sum of all the people in its connected component, weighted by social distance. We can thus see the social endorsement score as another instance of combining the rankings of different lists, where the "combining" here requires revising the scores in each list based on personalization. The difference from standard rank combination problems (see e.g. [Fag02]) is that exactly which users in the list contribute to the score is dependent on the querier. Standard algorithms for combining scores rely on the sorting of the inverted lists by static upper-bound scores. This is the case, for example, in the family of Threshold Algorithms [Fag02]. In the case of personalized scoring, it is not clear what kind of upper-bound could be used – a global (e.g., querier-independent) upper-bound would be too coarse since it overlooks the difference in behavior between users. A possible solution is to devise an adaptive algorithm that discovers friendships and resource endorsements during query evaluation and uses them to refine upper-bounds.

3.4 Refining Scores by Clustering

The social endorsement described in the previous section takes as given the fact that the score of a resource for a given query should depend upon the tagging activity of the members of a querier's explicitly-recognized community. The integrated approach allows the score to depend transitively on the impact of tagging, friendship links, resource hyperlinks, and resource content matching, but is still based on the notion of community given by explicit friendship information. One may be interested in using derived notions of affinity between users, creating either links between users or clusters of users based on common behavior. Derived links can be used as a substitute for explicit links in either an integrated or modular approach. User clusters can also play a role in gaining effectiveness by getting more personalized versions of algorithm parameters. In the integrated approach, this would mean replacing the various global damping parameters with multiple per-cluster weights. User clusters would also naturally fit in a modular approach, replacing the social-endorsement score by a per-cluster or per-term/cluster endorsement score for each resource. The definition of user clusters would help refine score upper-bounds. Since a querier only belongs to one cluster, the refined upper-bound of the cluster can be used for more effective pruning than a single score upper-bound per resource. The question of how many user clusters should be defined remains open.

Due to the large number of features of users, a possible approach to deriving user clusters is via machine learning. In this setting, training data corresponds to labeled resources that can be either inferred from click logs [XZC+04] or requested explicitly from users. The idea would be to cluster users based on their click behavior or on their explicit feedback on ranked results.

A promising aspect of a machine-learning approach is that it can exploit the feedback mechanisms already present in these sites to generate a significant amount of high-quality training data. We can easily imagine allowing end-users to evaluate rankings at query time. This would be particularly appealing to users in the context of online communities, due to their already-active participation in evaluating content. In particular, by making it ”easy” for users to mark a resource as good or bad, the user is only one click a way from providing that information, per resource. This is already enabled, in a limited way, by some systems such as the thumbs up/thumbs down feature in StumbleUpon [stu].

4 Conclusion

Social-content websites depend fundamentally on search functionality for their core value proposition. In this paper, we have outlined the relevance factors that must be combined for effective social-content search. We have presented an *integrated approach* in which any subset of these relevance factors can be mapped into a “random surfer” model and relevance calculated with a PageRank computation. In the face of feasibility and performance challenges with this technique, we have discussed the difficulties faced in adapting more traditional relevance computation techniques to the requirements of social-content search. The development of efficient search techniques capable of effective search in this domain is an important problem, just starting to be addressed by recent work [SBBW07, BWF+07, Cha07].

One challenge to such research is evaluating quality. Accepted standards of search quality typically involve carefully annotated example sets [tre, ine]. Providing the assessments on which these metrics are based is a tedious task, but one which is nevertheless necessary. Unfortunately, this task is even more difficult to implement in the context of personalized search in social content sites. To our knowledge, there is still no principled way to evaluate the quality of proposed relevance algorithms for search involving personalization or social distance. Research papers such as [SBBW07, BWF+07] rely on manual assessments done by individuals (e.g., paper authors and students). An important issue is how to compare quality of social ranking techniques across applications and research groups.

In the context of actual systems, however, there is a great potential for users to provide feedback to the system, since expressing opinions on topics of interest may be, along with finding interesting content, a key motivation for users visiting social content sites. One avenue to explore is the incorporation of techniques and interfaces built for collaborative filtering (see, for example, [HKTR04, KSS97]) to collect feedback from users, and thus evolve and tune relevance functions over time. A key issue in this process is how to *cluster* users to efficiently predict preferences across a variety of topics and content types.

References

[BBVW06] Klaus Berberich, Srikanta Bedathur, Michalis Vazirgiannis, and Gerhard Weikum. BuzzRank ... and the Trend is Your Friend. In *WWW ’06: Proceedings of the 15th international conference on World Wide Web*, 2006.

[BK07] Nilesh Bansal and Nick Koudas. Searching the Blogosphere. In *10th International Workshop on the Web and Databases (WebDB 2007)*, 2007.

[BP98] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30:107–117, 1998.

[BWF+07] Shenghua Bao, Xiaoyuan Wu, Ben Fei, Guirong Xue, Zhong Su, and Yong Yu. Optimizing Web Search using Social Annotations. In *WWW ’07: Proceedings of the 12th International World Wide Web Conference*, New York, NY, USA, 2007. ACM Press.

[CGS04] Yen-Yu Chen, Qingqing Gan, and Torsten Suel. Local Methods for Estimating PageRank Values. In *CIKM ’04: Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, 2004.

[Cha07] Soumen Chakrabarti. Dynamic Personalized Pagerank in Entity-relation Graphs. In *WWW ’07: Proceedings of the 16th International World Wide Web Conference*, pages 571–580, New York, NY, USA, 2007. ACM Press.

[Fag02] Ronald Fagin. Combining Fuzzy Information: an Overview. *SIGMOD Record*, 32(2):109–118, 2002.

[FR04] Dániel Fogaras and Balázs Rácz. Towards Scaling Fully Personalized PageRank. In *WAW '04*, volume 3243 of *LNCS*, pages 105–117, 2004.

[Hav02] Taher H. Haveliwala. Topic-sensitive PageRank. In *WWW '02: Proceedings of the 11th International World Wide Web Conference*, pages 517–526, New York, NY, USA, 2002. ACM Press.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[ine] Initiative for the Evaluation of XML Retrieval. http://inex.is.informatik.uni-duisburg.de/.

[JW03] Glen Jeh and Jennifer Widom. Scaling Personalized Web Search. In *WWW '03: Proceedings of the 12th International World Wide Web Conference*, pages 271–279, New York, NY, USA, 2003. ACM Press.

[Kle99] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46:604–632, 1999.

[KSS97] Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[MA00] David W. McDonald and Mark S. Ackerman. Expertise Recommender: A Flexible Recommendation System and Architecture. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work*, pages 231–240, New York, NY, USA, 2000. ACM Press.

[New] http://news.search.yahoo.com/.

[New00] M. E. J. Newman. Models of the Small World. *Journal of Statistical Physics*, 101(3-4):819–841, November 2000.

[RD02] Matthew Richardson and Pedro Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*, 2002.

[SBBW07] J. Stoyanovich, S. Bedathur, K. Berberich, and G. Weikum. Entityauthority: Semantically enriched graph-based authority propagation. In *10th International Workshop on the Web and Databases (WebDB 2007)*, 2007.

[SM83] Gerald Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[stu] `http://www.stumbleupon.com`.

[tre] Text REtrieval Conference. http://trec.nist.gov.

[WF94] Stanley Wasserman and Katherine Faust. *Social Network Analysis*. Cambridge University Press, Cambridge, 1994.

[WS98] Duncan Watts and S. H. Strogatz. Collective Dynamics of "small-world" Networks. *Nature*, 393:440–442, 1998.

[XZC$^+$04] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, Wensi Xi, and Weiguo Fan. Optimizing Web Search Using Web Click-through Data. In *CIKM'04*, pages 118–126, New York, NY, USA, 2004. ACM Press.

[ZPZ04] Li Zhang, Yue Pan, and Tong Zhang. Focused Named Entity Recognition using Machine Learning. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference*, 2004.

User-Centric Research Challenges in Community Information Management Systems

AnHai Doan[1], Philip Bohannon[2], Raghu Ramakrishnan[2]
Xiaoyong Chai[1], Pedro DeRose[1], Byron J. Gao[1], Warren Shen[1]
[1] University of Wisconsin-Madison, [2] Yahoo! Research

Abstract

In Cimple, a joint project between Wisconsin and Yahoo! Research, we are building systems that manage information for online communities. In this paper we discuss the fundamental roles users play in such systems, then the difficult user-centric research challenges raised by these roles, with respect to contributing to the system, accessing and using it, and leveraging the social interaction of users.

1 Introduction

In numerous online communities (e.g., those of database researchers, movie fans, and biologists) members often want to discover, query, and monitor relevant community information. *Community information management systems* (or *CIM systems* for short) aim to address such information needs [13]. First-generation CIM systems fall roughly into two classes: message boards and structured portals. In message-board systems (e.g., Usenet, Yahoo! Groups, DBworld), users exchange messages on active topics and the history of these messages provides a searchable repository of community knowledge. In contrast, portal systems include most enthusiast Web sites (e.g., *shakespeare-online.com*) and provide structured contents. While some portals (e.g. Citeseer [16]) have successfully presented automatically crawled content to users, most portal sites are maintained by a few system builders.

In Cimple, a joint project between Wisconsin and Yahoo! Research, we are developing techniques to build next-generation CIM systems [13]. Our first goal is to support *collaborative contribution and management* of a wide range of content (e.g., text, structured data, images). Our second goal is to minimize the information gathering load on community members by integrating *crawled Web content*. For example, in the DBLife prototype (see [12] and http://dblife.cs.wisc.edu), built as a part of the Cimple project, information of use to the database research community is crawled on a nightly basis. The challenge then is to integrate this data with the community-contributed text and structured data, while keeping quality high.

Several current projects are similar to Cimple in spirit, or share many of the goals. Examples include Impliance, MAFIA, and Avatar projects at IBM Almaden [8, 15, 23], BlogScope at the University of Toronto [7], BlogoCenter at UCLA [1], Dataspaces and PayGo at Google [19, 27], SHARQ and ORCHESTRA at the University of Pennsylvania [32, 9], Libra at MSR-Asia [28], related efforts at the University of Washington, MSR-Redmond [11, 17], Siemens Research [33], and many others (e.g., [6, 25], see also [14]). A key commonality underlying many of these projects is the *active and diverse roles* users play in building and using the

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

systems. Consequently, we believe that for these emerging "Web 2.0" projects, it is important to discuss which fundamental roles users play, and what user-centric challenges these roles entail.

In this paper we contribute to this broader discussion, drawing from our initial experience in Cimple. We begin by observing that CIM users often play three fundamental roles: *active contributors, information explorers*, and *social players*. First, CIM users often act as active contributors, editing and supplying the system with data, code, and domain knowledge. Second, CIM users often have ill-defined information needs (e.g., find interesting relationships between X and Y), or have precise information needs but do not know how to express them in structured query formats, or are too "lazy" to express them. Consequently, they often behave as information explorers. Finally, CIM users operate in a social context, in that they often interact with other users in the same community and that the CIM data captures many of such interactions.

We then discuss the user-centric challenges raised by the above observations. We consider in particular three key challenges: (1) how to make it easy for users to contribute data, code, and knowledge to the system, (2) how can users easily access and query the system, and move seamlessly from one query mode to another, and (3) how to motivate users to interact more, then capture and exploit such interactions. Finally, we discuss *reputation management, explanation*, and *undo*, capabilities that we believe are critical to address the above challenges.

2 The Fundamental Roles of CIM Users

We now briefly describe CIM systems, then the roles their users play. To build a CIM system, such as the one for the database research community, a builder (who is a community expert) deploys an initial generic system and supplies it with a set of relevant data sources (e.g., researcher homepages, DBworld mailing list, conference pages, etc.). The system then proceeds in three main steps [13]:

- **Crawl, extract, and integrate the data:** The system crawls the sources at regular intervals to obtain data pages, then extracts mentions of relevant entities from the pages. Example mentions include people names (e.g., "Jim Gray" and "J. N. Gray"), conference names, and paper titles. Next, it integrates these mentions into entities, and discovers relationships among them (e.g., "Jim Gray gave a talk at SIGMOD-04"), thus transforming the raw data into an entity-relationship (ER) data graph.

- **Provide user services over the data:** Next, the system provides a variety of user services over the ER data graph. For example, the system may create for each user entity X a *superhomepage* which contains *all* information about X that the system finds from the raw community data. Other example services include browsing, keyword and structured querying, and monitoring.

- **Mass collaboration:** Finally, the system solicits and leverages the feedback of community users to further improve and evolve. For example, the system may publish each user superhomepage (as described earlier) in wiki format, then allow users to correct and add information. A user may also suggest a new data source for the system to crawl. As yet another example, if the system infers both X and Y to be PC chairs of SIGMOD-04, a user may flag these inferences as incorrect, and supply the domain constraint that each SIGMOD conference has just one PC chair.

The Cimple project [13] (see also http://www.cs.wisc.edu/~anhai/projects/cimple) describes CIMS in more details. Within this project, to validate and drive CIMS research, we have also been building DBLife, a prototype system that manages the data of the database research community (see [12] and http://dblife.cs.wisc.edu). For CIM systems, we observe the following user roles.

Active Contributors: CIM users often want to contribute data, code, and knowledge to the system. In DBLife, for example, users sent us URLs of new data sources, voted on whether a picture claimed to represent a person

X is truly X, and inquired about supplying new codes for keyword search, mention disambiguation, among others.

User willingness to contribute of course has been observed in numerous Web 2.0 efforts. The amount of contribution has also been observed to follow a Zipfian distribution: a relatively small percentage of users contribute very actively, followed by a long tail of users who contribute little or nothing (e.g., see [5]). Our initial experience suggests that this will also hold for CIMS. Consequently, we roughly divide human participants of a CIM system into three categories: (a) *builders:* a small, perhaps 1-3 person, team which deploys and maintains the hardware and software (analogous to the DBA of an RDBMS), (b) *editors:* a core of perhaps 10-20 highly motivated persons who actively contribute to the system, and (c) *users:* the rest of the community. When there is no ambiguity, we use "users" to refer to all three categories.

While users are willing to contribute in many Web 2.0 efforts, as noted above, *in CIM contexts it is particularly important that they do so*. This is because, by nature, CIM data comes from multiple heterogeneous sources. They are often incomplete, only partially correct, and semantically ambiguous. Hence, it is vital that users contribute so that the data can be gradually cleaned, disambiguated, and augmented, especially in cases where it is very difficult for systems, but relatively easy for human users to make a decision. For example, it is very difficult for DBLife to decide that a picture of X is indeed X, whereas it would be easy for users who know X. As another example, a user can quickly tell the system that "Alon Halevy" and "Alon Levy" are the same person, saving it much effort in attempting to determine so. Note that this is in sharp contrast to RDBMS settings, where the data often has a closed-world well-defined semantics. Many data management settings outside RDBMS however have semantic problems (e.g., CIM, but also schema matching, data integration, data cleaning, dataspaces, and model management), and thus can significantly benefit from user participations.

Information Explorers: Recent work has addressed the needs of users who approach structured data sources with vague queries, by supporting keyword queries over structured data (e.g., [4, 21, 20, 18, 31]). Similarly, CIM users often have diverse, ill-defined information needs. Many times a CIM user does not yet know exactly what he or she wants (e.g., knowing only that he or she wants to find something interesting on topic X). Hence, the user will start with keyword search and browsing, in an exploratory fashion. This is especially true in scientific data management. Eventually the user may "zoom in" on a precise information need (e.g., find all papers on topic X that Y and Z wrote in 2004), at which point he or she may want to switch to a structured query interface. So a major problem is how to ensure a smooth transition across heterogeneous query and browsing interfaces, with minimal user effort.

Even if a CIM user starts with a precise information need, he or she often is too "lazy" to compose a structured (e.g., SQL) query, or simply does not know how to do it. In DBLife, for example, few users appear to be willing to take the effort to compose a structured query, or know how to compose a *syntactically correct* one. This is an acute problem, because it severely limits the utility of all the structured data that DBLife has extracted and integrated. Consequently, finding a way to allow lay or "lazy" users to ask structured queries in CIM contexts is very important, if we want to maximize the full utility of structured CIM data.

Social Players: CIM users operate within a community. They are often aware of and interact with other users, and such interactions are often captured in the data managed by a CIM system. Exploiting such data on social interaction can often significantly improve the quality of CIMS. For example, in DBLife, interaction in form of citations, paper review, tagging, etc. can help identify topic experts, and help improve ranking the results of keyword searches. Hence, a key challenge is how to encourage such social interactions, and how to capture and exploit them.

Finally, as we have alluded to several times, CIM users often vary significantly in their degree of motivation and technical expertise. While we expect that a relatively small core of users (e.g., the builders and editors, as described earlier) are highly motivated and technically literate, the vast majority of users will just want to use the system quickly if the need arises, then "move on with their lives". This exacerbates the user-centric challenges facing CIM systems, as we discuss next.

3 User-Centric Research Challenges

We now discuss the user-centric challenges, focusing in particular on user contribution, user services, and social interaction. Then we touch on reputation management, explanation, and undo, capabilities that are central to address the above challenges.

3.1 Effective User Contribution

Since user contribution is important for CIM, but the vast majority of users are reluctant to contribute, we must make it very easy for users to provide or modify system components. We focus on three main components: data, code, and domain knowledge.

Data: A user should be able to supply or edit any kind of data, using whichever user interface that he or she finds most convenient. The system then processes the data to its best ability. Example data include URL for a new data source, raw data pages (e.g., a page listing accepted SIGMOD papers), structured data, natural text, and tags, among others. Example user interfaces include form, GUI and wiki. Our Cimple experience suggests that wiki pages can provide a good baseline user interface, in that anything can be posted in wiki pages and can be easily edited. For instance, if DBLife displays user superhomepages in wiki format, then it is relatively easy for a user to correct and add information (especially natural text). Other interfaces can excel in certain cases. For example, a form interface is especially well suited for tagging data pieces with small text fragments.

In the above context, a major challenge is to translate user actions in an interface into actions over the underlying data. For example, conceptually a DBLife superhomepage describes a portion of the underlying ER data graph. Now suppose a user has revised a superhomepage (in wiki format). Then we must infer from the revised wiki page the exact sequence of actions the user intended to do over the ER data graph (e.g., remove a node, rename an edge, etc.). This inference is non-trivial because user edits often are ambiguous: the same edit can be mapped into multiple possible sequences of actions over the underlying data. Another challenge is that users often want to enter the data *together with some context information*. For example, when a user enters a page that contains a list of names, he or she may also want to say that these are the names of persons who are on the PC of SIGMOD-04.

Code: In practice, the code of a CIM system must often be tweaked to fine-tune the system performance. Today such tweaking is typically done by a small team of developers, incorporating suggestions from the members at large, in a slow and tedious process. This process can be improved markedly if we can open up certain parts of the code base for the multitude of members to edit.

To illustrate, consider extracting person names from the raw data pages. A common method is to start with a dictionary of names (e.g., "David Smith", "Michael Jones", etc.), perturb each name to generate variations (e.g., "D. Smith", "Smith, D"), then find occurrences of the variations in the raw pages. The method perturbs each name using the *same set of generic perturbation rules*. This often turns out to be insufficient. We found that when deployed in DBLife the method often had to be tweaked. It missed for example cases where a person X has an unusual nickname Y. Whenever this was pointed out to us by X or someone who knows X, someone on our development team would have to tweak the code, to add the nickname Y for X.

Clearly, allowing users to edit the code in such cases can drastically reduce the workload of the development team. Toward this goal, first we must make it very easy for users to edit the code. But it is unlikely that we can allow any user to edit code *directly*, as this can quickly result in corrupted code. A possible initial solution then is to (a) decompose the code into a sequence of tasks, (b) materialize the *output* of each task, then (c) allow users to edit only these outputs. For example, the name extractor described above can be decomposed into a sequence of two tasks: generating variations for each name, then finding occurrences of the variations. Thus, the name extractor should *materialize* the set of variations it generates for each name, and expose these materialized sets to the users, so that they can edit (e.g., add the nickname Y to the set for X). In general, we can identify certain "edit points" in the code, make sure that the code "materializes" these edit points, then expose them (e.g., via a wiki interface) to allow users to edit.

Another possible solution (to make it easy to edit code indirectly) is to define *multiple choices* at certain points in the code. The default code always takes the default choices. But users can select other choices, thereby changing the execution flow of the code. For example, consider a module that matches person names, e.g., deciding if "D. Smith" and "David Smith" refer to the same person. This module may use the default choice of always applying the *same* matching method m to *all* superhomepages. But it should also offer several other matching methods, and allow users to choose a particular matching method for a particular superhomepage, if the user so desires. Thus, while examining a superhomepage H, a user may decide to examine the code that matches names within H, then decide that a matching method m' (offered in the code) is actually more accurate for H. Consequently, the user tells the system (perhaps via a radio-button interface) that, whenever matching names within H, it should use the matching method m' instead of the default method m.

This last example illustrates the power of collaborative code editing in CIM settings. In such settings, the small team that writes the initial code simply cannot examine *all* superhomepages to write appropriate code for each superhomepage. But they can write the code in a way that makes it easy later for community users to adapt the code to the peculiarities of each superhomepage.

To address malicious code editing, an initial solution is to limit code editing to only "trusted" users (e.g., editors). Even in this case, distributed code editing is already very useful, as it spreads the workload over multiple people. It is also very important to develop an undo capability, so that undesired changes to the code can be undone easily. We discuss this capability in more details in Section 3.4.

Domain Knowledge: When a CIM user finds something incorrect, he or she often knows some domain knowledge that can be used to flag it as incorrect or to fix it. For example, when a user sees that the system claims both A and B chair SIGMOD-04, he or she may be able to supply the knowledge that "only one person chairs a SIGMOD conference". We found such cases commonly occur in DBLife. Thus, just as domain knowledge (e.g., integrity constraints) plays an important role in RDBMS, it also plays an important role in CIMS. Consequently, it is important to find ways to allow users to express a broad variety of domain knowledge. The key challenge is to make it very easy for lay users to do this.

A possible solution is to cast each piece of domain knowledge as a constraint Q *op* v, where Q is a query template formulated in a structured language (e.g, SQL), *op* refers to a predefined operator (e.g., =, $<$, etc.), and v is a value. The user then interacts with the system to construct Q, then select *op* and v. For example, to express the constraint "only one person chairs a SIGMOD conference", the user constructs a template Q that finds the number of chairs of any given SIGMOD conference, then sets *op* to be $=$, and v to be 1. Another solution is for the system to solicit domain knowledge from the user. For example, while constructing a profile of a typical database researcher, a system may infer a constraint such as "no database researcher has published four or more SIGMOD papers in a year". It can then ask users to verify this constraint with answer "yes" or "no".

3.2 Effective User Services

As discussed earlier, CIM users often have ill-defined information needs, or do not know how to formulate the need in a structured query, or are too "lazy" to do so. Within this context, we must make it very easy for users to access and utilize the system. We now discuss the challenges in doing so, focusing on querying, context-sensitive services, and system access.

Querying: A user should be able to query the system using whichever query mode he or she finds most convenient, and should be able to switch seamlessly among them, with minimal effort. Example query interfaces include keyword search, GUI search, and structured querying. How to query effectively in each of these modes remains a major challenge. For example, while much work has addressed "plain-vanilla" keyword search (which returns a ranked list of data pages), no satisfactory solution exists today that can be adapted to work effectively, with minimal tuning, in a CIM domain. Similarly, much work has addressed keyword search over structured data, but no consensus has emerged on the most effective solution. Furthermore, how to execute structured queries over extracted structured data has received relatively little attention (with some exceptions [11, 22]). This last problem is difficult because the extracted structured data is often incomplete and imprecise.

Another major challenge is how to make smooth transition from one query mode to another. To move from a less structured query mode to a more structured one, a common solution is to interact with the user to refine the query [23, 26]. In the Avatar project [23], for example, when a user asks a keyword query "tom phone" over a corpus of emails, the system returns a ranked list of emails that contain these words. But it also provides an opportunity for the user to move to more structured querying, by asking if the user means to find emails that contain the phone number of Tom, or to find emails that come from Tom and contain the word "phone". There are often numerous possible structured-query interpretations for a keyword query. Hence a key difficulty facing this solution is how to select only the most likely interpretations, to show the user. User modeling (e.g., [3]) may help facilitate this selection. To move from a more structured query mode to a less structured one (e.g., when the more structured query does not produce any result and hence must be "relaxed", or when it cannot be executed over a text corpus), a common solution is to "collapse" the structured query, for example, into a set of keywords [30, 24]. The key issue is then how to select a good set of keywords.

Yet another major challenge is that once a CIM system has compiled a structured database, how can it enable users to easily pose structured queries over the database? For example, a user may want to know the average number of citations per paper for a particular researcher X. Clearly the system cannot expect that most users will be able to write a structured query (e.g., in SQL) expressing this information need. A possible solution is then for the system to interact with the user in a GUI fashion to construct a structured query.

Another possible solution is to generate form interfaces that capture the types of structured queries that we expect users will commonly ask. This is also the preferred approach for today RDBMS applications (e.g., $amazon.com$ provides a small set of form interfaces for users to query about books). CIM users however often have ill-defined and exploratory information needs (as discussed in Section 2). Consequently they often want to ask a far wider and more unpredictable range of structured queries. Thus, the CIM system may have to generate a very large number of form interfaces. Hence, for this approach to work, the system must be able to index these interfaces, and then return the most relevant ones, given a user's keyword query.

Context-Sensitive Services: To minimize user efforts and maximize their utilization of a CIM system, the system should provide context-sensitive services. For example, when the user accesses a page that contains publications, the system can consider all actions (querying, monitoring, etc.) that a user may want to do with those publications, then offer to execute those actions. These offers can be listed, e.g., on the right side of the page, similar to the way advertisements are displayed in search engine result pages. The key challenge here is to decide on which services to offer that would maximize users' utilization of the system, a challenge that is akin to deciding which advertisements to display in a search result page.

Easy Access to the System: Finally, we cannot just rely on users going to the system frontpage to ask queries or to browse. Most users today suffer from information overload. It is likely that they will just use a major search engine (e.g., Google, Yahoo) most of the time to search for information, an observation also made by [26]. Hence, it is very important that we "open up" a CIM system for major search engines to crawl and index, so that when a user asks a keyword query that can potentially be answered by the system, then the search engine will return a page of the system in the top few results. The key challenges then are (a) how to maximize the chance that search engines will place a CIM system page high in the ranked list, if by accessing that page, the user can fulfill his or her information need, and (b) once the user has accessed the page, how to enable the user to quickly express his or her information need, then answer it.

3.3 Encouraging, Capturing, and Exploiting Social Interactions

So far we have discussed CIM users in isolation. But a distinguishing characteristic of CIM settings is that the users form a community: they often interact with one another, and such interactions are often captured in the data. Hence, we should design CIM systems such that they encourage such social interactions, capture them, and exploit them.

To encourage social interactions, CIM systems can employ a plethora of social tools such as those that allow users to tag, blog, comment, bookmark, form mailing lists, etc. And indeed many current social networking systems deploy such tools. The main problem is that we simply do not know when a particular tool will work (in that many users will use it). Hence, we foresee two major challenges. The first challenge is to develop more social tools, on the ground that expanding the tool collection makes it more likely that users will find something they like, and thus initiating more social interaction. The second challenge is to develop a mechanism to systematically deploy combinations of social tools in a CIM setting, evaluate their effectiveness in encouraging user participation, and then retain and improve the best ones.

Many CIM users also interact *outside* the system, but traces of such interactions are often captured in the raw data. For example, if X appears on the PC of a workshop organized by Y, then it is likely that X and Y have exchanged emails and are sharing some common interests. Hence, another challenge is to mine such social interactions from the raw community data. While mining social interactions is not a new topic, a distinguishing aspect of CIM settings is the abundance of *temporal data*. CIM systems crawl and archive community data over time (e.g., DBLife has crawled and archived the data of the database research community over the past 2.5 years). Exploiting the temporal aspect of this data may allow us to infer social interactions and their strengths more accurately.

Once social interactions have been captured or inferred, they can be exploited for many purposes, such as enhancing keyword search, identifying experts, finding emerging hot trends, viral marketing of ideas and services, among others. This has been a very active area of research (e.g., see the proceedings of recent WWW, KDD, database, and AI conferences). In CIM contexts, since feeding data into the system and querying it pose major difficulties (as discussed in Sections 3.1 and 3.2), an important challenge is to find out how to exploit social interactions to address these difficulties.

3.4 The Enablers: Reputation Management, Explanation Generation, and Undo

We have discussed user contribution, user services, and social interaction. These challenges share a set of core problems, and hence it is important that we develop effective solutions to these problems. We consider in particular reputation management, explanation generation, and undo.

Reputation management means knowing how much to trust any user X and to manage X's contributions to the CIM system. Much work has addressed reputation management (e.g., [2, 29]), but no consensus has emerged on the best method, and it is unlikely that a single silver bullet exists. Hence, like the case for social tools, an important challenge is to develop solutions that deploy reputation management tools, evaluate them, and retain and improve the best ones.

Explanation generation means that the system can explain to a user why a particular inference is made (e.g., why X is a PC member of conference Y) or *not* made (e.g., why didn't the system infer that Z is also a PC member of Y). We found that users asked many such questions in the DBLife context, either because they simply wanted to know, or because they used the explanations to decide on how much to trust the inference made by the system. We ourselves also often asked such questions for debugging purposes. Hence, providing explanations is important for the effective development and utilization of CIM systems. Further, showing explanations also often allows better user corrections. For example, if a user only says "this output is wrong", the system has to infer which operator or datum involved in producing that output is the culprit. However, if the user can see an explanation, he or she may be able to pinpoint the error for the system.

Providing explanations on why a particular inference is made can utilize lineage (a.k.a. provenance [10, 34]) maintained by the system. The problem of providing explanations on why a particular inference is *not* made appears to be far harder, and has received little attention.

Finally, the undo capability allows users to roll the system back to a previous state. This capability is absolutely critical. As one user explained to us "without knowing that I can undo, I will not be willing to experiment with the features that the system provides". As Wikipedia demonstrates, undo is also important for managing malicious users. To enable this capability, a CIM system must log *everything*, including all user interactions. Then, the system must decide how much to allow users to undo. The problem is that if the system allows users to undo deep into the "past", it must limit concurrent editing of users, or risks losing user edits that build on a "transaction" that is later undone. How to strike the right balance here is a difficult question.

4 Concluding Remarks

As our field expands beyond managing structured data, to consider unstructured data in "Web 2.0" contexts, it is important that we discuss how the role of users has fundamentally changed in the new contexts, and what user-centric challenges those changes entail.

In this paper we have contributed to this broader discussion, drawing from our initial experience in the Cimple project on community information management systems. We described how users of such systems often act as active contributors, information explorers, and social players. For the role of active contributors, the key challenge is to enable users to supply or edit any kind of data, code, and domain knowledge, using whichever user interfaces they find most convenient. For the role of information explorers, the key challenge is to enable users to query the system using whichever query mode they find most convenient, and to switch seamlessly between the query modes with minimal effort. For the role of social players, the key challenge is to develop a broad range of social tools and mechanisms to select the most effective tools. Finally, we made the case that reputation management, explanation generation, and undo are critical in addressing the above challenges.

References

[1] http://oak.cs.ucla.edu/blogocenter.

[2] B. Adler and L. Alfaro. A content-driven reputation system for Wikipedia. In *Proc. of WWW-07*, 2007.

[3] E. Agichtein. Web information extraction and user modeling: towards closing the gap. *IEEE Data Engineering Bulletin*, 28(4), 2005.

[4] S. Agrawal, S. Chaudhuri, and G. Das. DBexplorer: A system for keyword search over relational databases. In *Proc. of ICDE-02*, 2002.

[5] R. Almeida, B. Mozafari, and J. Cho. On the evolution of Wikipedia. In *Proc. of the Int. Conf. on Weblogs and Social Media*, 2007.

[6] S. Amer-Yahia. A database solution to search 2.0 (keynote talk). In *Proc. of WebDB-07*, 2007.

[7] N. Bansal and N. Koudas. Blogscope: Spatio-temporal analysis of the blogosphere. In *Proc. of WWW-07*, 2007.

[8] B. Bhattacharjee, J. Glider, R. Golding, G. Lohman, V. Markl, H. Pirahesh, J. Rao, R. Rees, and G. Swart. Impliance: A next generation information management appliance. In *CIDR*, 2007.

[9] S. Boulakia, O. Biton, S. Davidson, and C. Froidevaux. Bioguidesrs: Querying multiple sources with a user-centric perspective. In *Bioinformatics*, 2007.

[10] P. Buneman and W. Tan. Provenance in databases (tutorial). In *Proc. of SIGMOD-07*, 2007.

[11] M. Cafarella, C. Re, D. Suciu, O. Etzioni, and M. Banko. Structured querying of Web text data: A technical challenge. In *Proc. of CIDR-07*, 2007.

[12] P. DeRose, W. Shen, F. Chen, Y. Lee, and D. Burdick. DBLife: A community information management platform for the database research community (demo). In *Proc. of CIDR-07*, 2007.

[13] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. *IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases*, 29(1), 2006.

[14] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing information extraction (tutorial). In *Proc. of SIGMOD-06*, 2006.

[15] Mehmet Altinel et. al. Mafia: A mashup fabric for intranet applications (demo). In *Proc. of VLDB-07*, 2007.

[16] C. Giles, K. Bollacker, and S. Lawrence. Citeseer: an automatic citation indexing system. In *Proc. of DL-98*, 1998.

[17] M. Gubanov and P. Bernstein. Structural text search and comparison using automatically extracted schema. In *Proc. of WebDB-06*, 2006.

[18] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRank: Ranked keyword search over xml documents. In *Proc. of SIGMOD-03*, 2003.

[19] A. Halevy, M. Franklin, and D. Maier. Principles of dataspace systems (invited paper). In *Proc. of PODS-06*, 2006.

[20] V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword search in relational databases. In *Proc. of VLDB-02*, 2002.

[21] A. Hulgeri and C. Nakhe. Keyword searching and browsing in databases using BANKS. In *Proc. of ICDE-02*, 2002.

[22] A. Jain, A. Doan, and L. Gravano. SQL queries over unstructured text databases. In *Proc. of ICDE-07 (poster)*, 2007.

[23] R. Krishnamurthy, S. Raghavan, J. Thathachar, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. *IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases*, 29(1), 2006.

[24] J. Liu, X. Dong, and A. Halevy. Answering structured queries on unstructured data. In *Proc. of WebDB-06*, 2006.

[25] J. Luxenburger and G. Weikum. Exploiting community behavior for enhanced link analysis and web search. In *Proc. of WebDB-06*, 2006.

[26] J. Madhavan, A. Halevy, S. Cohen, X. Dong, S. Jeffery, D. Ko, and C. Yu. Structured data meets the Web: A few observations. *IEEE Data Engineering Bulletin*, 29(4), 2006.

[27] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-scale data integration: You can only afford to pay as you go. In *Proc. of CIDR-07*, 2007.

[28] Z. Nie, J. Wen, and W. Ma. Object-level vertical search. In *Proc. of CIDR-07*, 2007.

[29] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[30] P. Roy, M. Mohania, B. Bamba, and S. Raman. Toward automatic association of relevant unstructured content with structured query results. In *Proc. of CIKM-05*, 2005.

[31] M. Sayyadian, A. Doan, and L. Gravano. Efficient keyword search over heterogeneous relational databases. In *Proc. of ICDE*, 2007.

[32] N. Taylor and Z. Ives. Reconciling changes while tolerating disagreement in collaborative data sharing. In *Proc. of SIGMOD-06*, 2006.

[33] F. Wang, C. Rabsch, P. Kling, P. Liu, and P. John. Web-based collaborative information integration for scientific research. In *Proc. of ICDE-07*, 2007.

[34] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proc. of CIDR-05*, 2005.

Increasing the Predictive Power of Affiliation Networks

Lisa Singh
Computer Science Dept.
Georgetown University
Washington, DC, USA
singh@cs.georgetown.edu

Lise Getoor
Computer Science Dept.
University of Maryland
College Park, MD, USA
getoor@cs.umd.edu

Abstract

Scale is often an issue when attempting to understand and analyze large social networks. As the size of the network increases, it is harder to make sense of the network, and it is computationally costly to manipulate and maintain. Here we investigate methods for pruning social networks by determining the most relevant relationships in a social network. We measure importance in terms of predictive accuracy on a set of target attributes of social network groups. Our goal is to create a pruned network that models the most informative affiliations and relationships. We present methods for pruning networks based on both structural properties and descriptive attributes. These pruning approaches can be used to decrease the expense of constructing social networks for analysis by reducing the number of relationships that need to be investigated and as a data reduction approach for approximating larger graphs or visualizing large graphs. We demonstrate our method on a network of NASDAQ and NYSE business executives and on a bibliographic network describing publications and authors and show that structural and descriptive pruning increase the predictive power of affiliation networks when compared to random pruning.

1 Introduction

A social network describes a set of actors (e.g., persons, organizations) linked together by a set of social relationships (e.g., friendship, transfer of funds, overlapping membership). Social networks are commonly represented as a graph, in which the nodes represent actors and edges represent relationships. Examples of social networks include online communication networks, disease transmission networks, and bibliographic citation networks. There is a growing interest in methods for understanding, mining, and discovering predictive patterns in social networks.

An *affiliation network* is a special kind of social network in which there are two kinds of entities, actors and events, and there is a participation relationship which relates them. Affiliation networks are commonly represented as bipartite graphs, in which there are two kinds of nodes, representing actors and events, and edges link actors to events. Examples of affiliation networks include: 1) corporate board memberships, where the actors are executives, the events correspond to different company boards, and the links indicate which executives serve on which company boards; 2) author collaboration networks, where the actors are authors, the events are

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

papers, and the links indicate co-authors of papers; and 3) congressional voting records, where the actors are the congressional members, the events are the bills, and the links represent the supporters for a bill.

A social network has both structural properties and descriptive attributes. The structural properties are determined by the graph structure of the network. Examples include the density of the graph, the average degree of nodes in the graph, the geodesic distance in the graph, the number of cliques in the graph, etc. In addition to structural properties, actors, events and relationships often have associated descriptive attributes containing features specific to the social context of the network. These are typically represented as attributes of the nodes or edges. For example, a corporate board social network may contain descriptive attributes representing the job function and age of a board member. A disease transmission social network may contain descriptive attributes representing the location of person's home and date of disease discovery.

Recent literature in the network science community has focused on understanding the structural properties of social networks and the construction of models for generating networks which have certain structure characteristics (degree distribution, small-world effects, etc.). Computer scientists are mining social networks based on these structural properties of networks. However, developing methods which combine network structure and descriptive attributes are necessary for accurate predictive modeling.

Predictive modeling can also be used to study approaches for compressing the representation of a social network, while maintaining its predictive accuracy. In the past, the social networks as studied in sociology tended to be relatively small, often with only tens of nodes. However, given the great increase in ability to both gather and process data, the social networks being analyzed today can be quite large. Because the data used to describe the network may not originally have been collected for the purpose of social network analysis, the data may contain irrelevant, redundant or noisy information. Noisy and redundant information can make networks difficult to interpret. Automatic techniques for identifying relevant aspects of the social networks can help improve computational efficiency and may at the same time improve understandability. Furthermore, since recording changes to a social network and maintaining consistency can be expensive, some applications can benefit from minimizing the amount of information stored.

In this paper, we begin by giving an overview of some of the representational issues related to social networks, especially affiliation networks. Next, we describe different pruning strategies for social networks. Our aim is to find compressed networks that maintain predictive and descriptive quality. Here we measure the compression in terms of the description length of the network and we measure the quality by measuring the predictive accuracy for the event attribute classifier built from the compressed network. We have evaluated our pruning methods on two real-world data sets. One is a network of NASDAQ and NYSE business firm executives and board members. The second is a bibliographic network describing publications and authors. We have found that we can achieve significant compression without sacrificing (and in some cases improving) predictive accuracy. This paper extends the work introduced in [17].

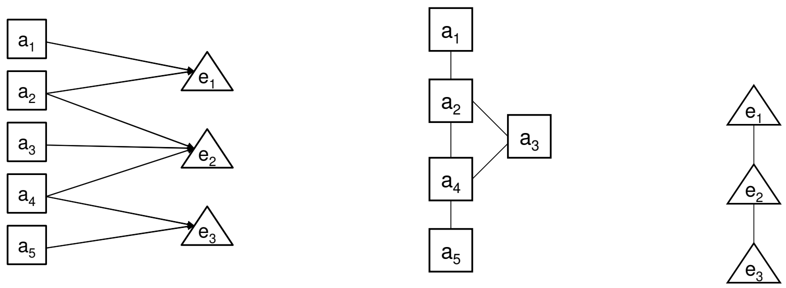
2 Affiliation Networks

Definition 1: An *affiliation network* N consists of a set of actors A, linked via a set of relationships R to a set of events E, $N = A, R, E$, where

$$\begin{aligned} A &= \{a_1, \ldots, a_n\}, \\ E &= \{e_1, \ldots, e_m\}, \\ R &= \{r_{ij}\}, \text{ where } r_{ij} \text{ denotes actor } a_i \text{ participates in event } e_j, \end{aligned}$$

and n is the number of actors and m is the number of events.

An affiliation network may be represented using many different graph structures. The most common representation for affiliation networks is as a bipartite graph, which we will call an *actor-event* graph, AE. In



(a) (b) (c)

Figure 1: (a) A simple affiliation network with actors a_1, a_2, a_3, a_4 and a_5 and events e_1, e_2 and e_3 (b) The co-membership graph for the affiliation network (c) The event overlap graph for the network.

this representation, there are two different node types representing actors and events. Networks with two node types are called *two-mode* or *bi-modal*. Figure 1(a) shows a small example of a two-mode actor-event node graph. The squares in the figure represent actors and the triangles represent events. The membership relations are highlighted in this graph structure.

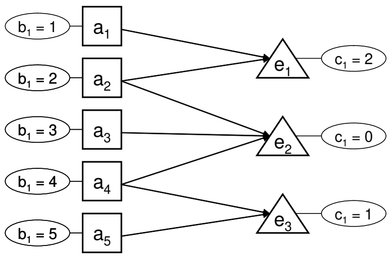
There are several useful projections of the actor-event graph. To focus on actors, one can perform a unipartite projection of the actors on the two-mode affiliation graph. The resulting network is a *single-mode* or *uni-modal* network, where we have a single object type and a single edge type. Representing an affiliation network in this way results in what is referred to as the *co-membership* graph, CM. The co-membership graph has a node for each actor, and an edge between actors who participate in the same event. Similarly, to focus on events, one can projection the actor-event graph onto the events. This results in what is called an *event overlap* graph, EO. It also contains a single node type and a single edge type. In the event overlap graph, the emphasis is on the connections among events. This graph has a node for each event, and an edge between events that share a common actor. Figure 1(b) shows the co-membership graph corresponding to the actor-event graph in Figure 1(a), and Figure 1(c) shows the event overlap graph corresponding to the same actor-event graph.

In addition to the nodes and edges themselves, the nodes and edges in the affiliation network can have descriptive attributes or features associated with them. Figure 2(a) shows the affiliation graph along with descriptive attributes for the actors and events (shown in ovals). In a corporate board social network, executives may have attributes such as education level, academic degree and age, companies may have attributes describing the corporation such as industry, sector, stock exchange and share price, and the serves-on-board relation may have attributes describing the relationship between the corporation and the executive such as position on the board and length of tenure on the board.

It is straight-forward to represent an affiliation network in relational algebra. We introduce the relations $A(Id_A, B_1, \ldots, B_k)$,$E(Id_E, C_1, \ldots, C_l)$, and $R(Id_A, Id_E, D_1, \ldots, D_m)$, representing the actors, the events, and the participation relations of a network. Here the Id_A, Id_E, and (Id_A, Id_E) are primary keys and the B_i, C_j and D_k are descriptive attributes for the relations A, E and R, respectively.

3 Prediction in Social Networks

Our goal is to develop principled approaches to compressing and pruning social networks. Our approach is to determine which portions of the network can be removed while minimizing information loss. Let $N = (A, E, R)$ be ther original network and $N' = (A', E', R')$ be the pruned network (we will describe how we construct the pruned network shortly). We begin by describing the predictive accuracy measure used to assess the performance



(a)

Id_E	c_1	$Max(b_1)$	$Avg(c_1)$
1	2	2	0
2	0	4	1.5
3	1	5	0

(b)

Figure 2: (a) The affiliation graph with descriptive attributes for the actors and events shown in ovals. (b) The constructed attributes for the events.

of different pruning approaches.

Here, we will focus on maximizing our predictive accuracy on the event attributes. For ease of exposition, we will assume we are attempting to maximize the predictive accuracy for a single event attribute $E.C_i$, based on attributes of related actors found using the co-membership information and based on attributes of related events found using the event overlap information. The idea is to construct a classifier, using local neighborhood information, to predict $E.C_i$. Now it is easy to see the difficulty with this setup. Each event may have a different number of related actors and a different number of related events, so how can we construct features to use in our classifier?

We solve this problem by computing an aggregate over the set of related actors and over the set of events. Aggregation is a common technique used to construct feature vectors in relational domains [11, 15]. Here we assume some set of aggregates is associated with each attribute. For the actor attributes $\{B_1, \ldots, B_k\}$, we have associated aggregate operators $\{a_{B_1}, \ldots a_{B_k}\}$ and for the event attributes $\{C_1, \ldots, C_l\}$, we have associated aggregate operators $\{a_{C_1}, \ldots, a_{C_l}\}$,

We begin by computing the aggregates over the set of related actors: [1]

$$AA(Id_E, A_{B_1}, \ldots, A_{B_k}) = \\ \gamma_{Id_E, a_{B_1}(B_1), \ldots, a_{B_k}(B_k)}(R \underset{R.Id_A = A.Id_A}{\bowtie} A)$$

which we call AA for aggregates over actors. Next we compute aggregates constructed from the related events:

$$AE(Id_E, A_{C_1}, \ldots, A_{C_l}) = \\ \gamma_{Id_E, a_{C_1}(C_1), \ldots, a_{C_l}(C_l)}(EO \underset{EO.R.Id_E = E.Id_E}{\bowtie} E)$$

which we call AE for aggregates over events.

We can combine these relations with the event relation E to create a set P_E containing both constructed features and event attributes.

$$P_E = E \underset{E.Id_E = AA.Id_E}{\bowtie} AA \underset{E.Id_E = AE.Id_E}{\bowtie} AE$$

[1]Recall that γ is the grouping operator in relational algebra [6].

We will use the constructed features to predict event attributes.
Example: Consider the affiliation network with descriptive attributes shown in Figure 2(a). Suppose that the aggregate operator that we use for B_1 is maximum and the aggregate operator that we use for C_1 is average. The constructed table showing the aggregates that will be used to build our classifier is shown in Figure 2(b).

The above describes in a generic way how we find the features from which we will predict event attributes. In order to actually make a prediction, we will need to first learn a classifier. Here we do not do anything out of the ordinary; we construct an appropriate training set from an observed social network. The constructed training set can be used by any supervised learning method to learn a classifier F, which predicts the value of E.C based on $\{A_{B_1}, \ldots, A_{B_k}, A_{C_1}, \ldots, A_{C_l}\}$.

We compare the classifier F_N constructed from the original social network $N = \{A, E, R\}$ with the classifier $F_{N'}$ constructed from a pruned social network $N' = \{A', E', R'\}$. We compare both accuracy on the training sets and, more importantly, accuracy on test sets. Accuracy on the training set measures how well the classifiers are able to fit the existing network. Accuracy on the test set measures how well the classifiers are able to generalize. Our goal is to find pruned networks that are both compact and accurate on both sets.

4 Pruning Techniques

Next we describe different pruning strategies. We consider two categories of operations. The first involves removing edges from the affiliation network. The second involves removing actors (and incident edges) from the affiliation network. We can use different techniques for pruning a network. The three techniques of interest to us are: 1) pruning based on structural properties, 2) pruning based on descriptive attribute values, and 3) pruning based on random sampling.

Structural Pruning Structural network properties or measurements involve evaluating the location of actors in a social network. Measuring the network location involves finding the centrality of a node. Structural measures have traditionally been used to identify prominent or important nodes in a social network. Two well known centrality measures are *degree* and *betweenness*. The degree of a node is defined as the number of direct connections a node has to other nodes in the network. The nodes with the most connections are considered the most active nodes in the network. They are referred to as the connectors or the *hubs* in the network. Betweenness of a node corresponds to the number of shortest paths going through the node. Nodes with high betweenness are referred to as *brokers*. A variation of this that is appropriate for affiliation networks is the number of cliques a node connects. This allows us to identify nodes that connect one group of actors to another group of actors. In traditional uni-mode networks, this could be a node that links two clusters that it does not participate in. It acts as a bridge between these clusters. In affiliation networks, this measure identifies nodes that participate concurrently in multiple events. These brokers are boundary spanners that have access to information flow in multiple clusters. They tend to have great influence in the network [19].

Therefore, when pruning based on structure, we will be interested in removing actors that are not hubs and/or brokers from the network.

Descriptive Attribute-based Pruning Another pruning technique of interest involves pruning based on descriptive attributes. We prune edges by selecting on attributes D_j of the R relation,

$$R' = \sigma_{R.D_j=d_j}(R),$$

where d_j is some constant attribute value. In other words, we will remove edges from our graph based on values for D_j. We look at both the case where we keep *only* edges with value d_j for D_j, and also the case where we keep all edges *except* edges with value d_j. Pruning edges may result in pruning both actor and event nodes if after pruning there are no edges connecting them to the network.

In addition, we prune actors by selecting on attributes B_j of actor relation A,

$$A' = \sigma_{A.B_j=b_j}(A),$$

where b_j is some constant attribute value. Pruning actors also results in a reduction in the number of edges, since we drop any edges to non-selected actors.

Random Sampling Finally, as a baseline, we compare pruning based on random sampling. This involves maintaining only a random sample of the actor population for analysis. Random sampling is a traditional statistical approach to approximating large graph structures.

Compression It is important to quantify the compression achieved by pruning. We use a relatively generic measure, the description length of the graph,

$$DL(N) = \log(|A|) + \log(|E|) + |R|(\log(|A|)\log(|E|))$$

where the logs are base two. $DL(N)$ is the number of bits required to represent the network. We need the first two terms to describe the number of actors and the number of events and the final term is the number of bits required to represent the edges.

5 Experimental Results

In this section we evaluate the degree of compression and the predictive accuracy of different pruning approaches.

5.1 Data Sets

We analyzed two affiliation networks. The first data set, the Executive Corporation Network (ECN), contains information about executives of companies that are traded on the NASDAQ and the NYSE. The executives serve on the Board of Directors for one or more of the companies in the data set. This data was collected from the Reuter's market data website (yahoo.mulexinvestor.com) in January 2004. There are 66,134 executives and 5384 companies (3284 NASDAQ and 2100 NYSE). The executives are the actors in the ECN, the companies are the events and board membership is the connecting relationship between the actor nodes and the event nodes. The relational schema is:

- A = Executive(exec_id, exec_name, age, education_level)
- E = Company(co_id, co_name, stock_exchange, sector, stock_price)
- R = BoardMembership(exec_id, co_id, officer_position, join_date)

The average board size is 14, the average number of boards an officer is on is 1.14, the number of officers serving on multiple boards is 6544, and the average number of boards these officer are on is 2.4. We attempt predicting two attributes, *stock_exchange* and *sector*. A sector is a coarse grouping of industries of the companies, e.g., telecommunications and health care. When pruning on descriptive attributes, we consider attributes of both the Executive relation and the BoardMembership relation. One example is *officer_position*, e.g., CEO, President, Treasurer and Director.

The second data set, the Author Publication Network (APN), contains information about publications and their authors. This data set was created using a portion of the ACM SIGMOD anthology in 2004. We focused on a subset of the periodicals and authors where there was at least one reference to the publication. In the final data set we analyzed, there were 13,070 authors and 16,287 publications.

The authors are the actors in the APN and the publications are the events. Paper authorship is the connecting relationship. The relational schema is:

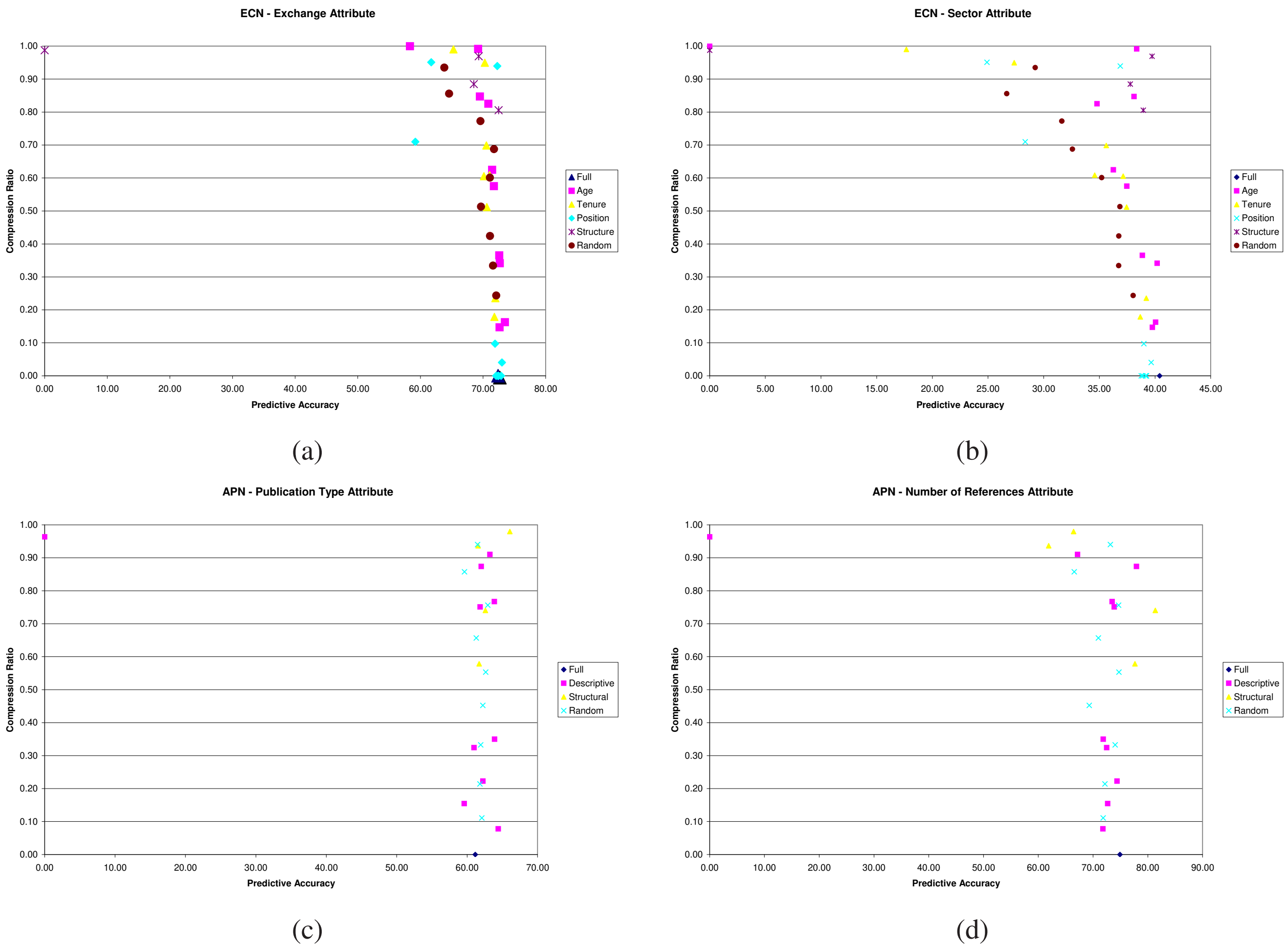


Figure 3: Comparisons of compression vs. accuracy for a variety of network pruning strategies for a) ECN exchange b) ECN sector c) APN pulbication type and d) APN number of references.

- A = Author(author_id, author_name, affiliation, number_of_publications)
- E = Publication(pub_id, pub_type, pub_date, number_of_references, number_of_citations)
- R = PaperAuthorship(author_id, pub_id)

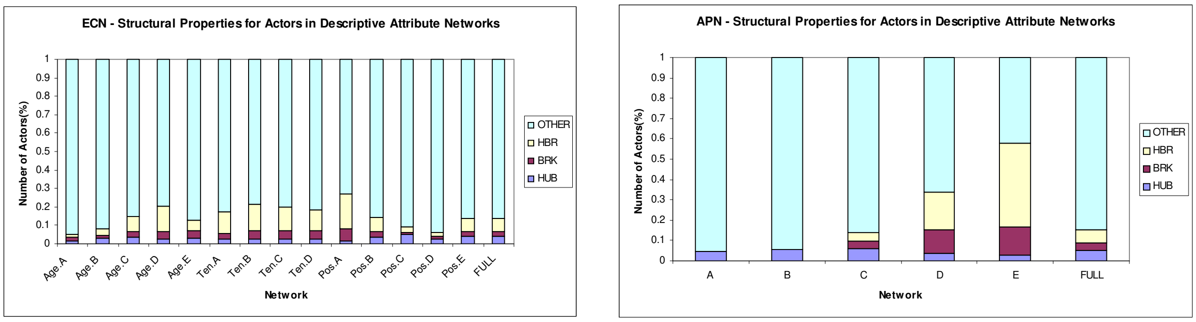
The average number of authors per publication is 2.4 and the average number of publications per author is 2.9. For APN, we predicted the two event attributes *pub_type* and *number_of_references* (to publication).

5.2 Accuracy and Compression Results

Our goal is to find small networks that can accurately predict event attributes. We compare the following affiliation networks:

- no pruning (**full**)
- descriptive attribute pruning (**descriptive**)
- pruning based on hubs and/or brokers (**structural**)
- random sampling (**random**)

We built event-attribute classifiers from the networks as described in Section 3. For categorical aggregate attributes, we calculated the mode of the neighboring event values, and for numeric aggregate attributes, we



(a) (b)

Figure 4: The structural characteristics of actors in different prunings for a) ECN and b) APN.

calculated the minimum, maximum and average of the neighboring event values. Once the predictive models have been generated, we evaluate the predictive accuracy of the complete network and the different pruned networks. We also compare the compression rations in terms of descriptive length, $DL(G)$. The classifiers were then created using WEKA. We tested a range of classification algorithms including decision trees, Naive Bayes, and support vector machines (SVMs). The results were relatively consistent across classifiers; due to space constraints, here we present results only for SVMs using five-fold cross validation.

When constructing our feature vector, we constructed aggregates for the following ECN actor and event attributes: stock exchange, industry, sector, number of officers on a board, number of advanced degrees on a board and officer age of a board. We evaluated three descriptive prunings. The first two descriptive prunings, *position* and *tenure*, involve removing edges from our affiliation graph for executives based on the attributes *BoardMembership.officer_position* and *BoardMembership.join_date*. For example, one pruning of *BoardMembership.officer_position* keeps only edges of CEOs and removes all other membership edges from the network. The third descriptive pruning involves removing actors based on age.

To group attribute values, we binned numeric attributes and we abstracted categorical attributes. Binning for each descriptive attribute used for pruning was created based on maintaining approximate equal size buckets or based on sematically interpretable abstractions. For both our networks, the binnings resulted in four to five bins for each attribute. For example, the attribute bands for *BoardMembership.officer_position* are as follows:

- A - Chairman of the Board
- B - Executive Officer (CEO, President, COO, etc.)
- C - Senior Officer (VP, Sr. VP, Comptroller, etc.)
- D - Board Officer (Treasurer, Secretary, etc.)
- E - Director

For the APN, we used the attribute *Author.number_of_publications* for descriptive pruning.

As mentioned earlier, descriptive attribute pruning has one of two interpretations for an attribute B with attribute value c: 1) maintain *only* actors with $B = c$ (**only**) and 2) maintain all actors *except* where $B = c$ (**except**). We evaluated pruning on every descriptive attribute value for each descriptive pruning category.

For structural pruning, we tested four cases: maintaining only actors who are hubs, (**HUB**), maintaining only actors who are brokers, (**BRK**), maintaining only actors who are both hubs and brokers, (**BOTH**), and maintaining only actors who are hubs or brokers. (**HBK**). Finally, for random pruning, we compared results on random samples for 9 different sample sizes (between 10% and 90% of the actors in the network).

Figure 3 shows compression versus predictive accuracy for two different attributes in each data set. The right upper corner represents the 'best' networks in terms of compression and predictive accuracy. Figure 3(a) shows results for predicting the ECN *exchange* attribute. The classifier built using the full network achieves an accuracy of 72.4%. The best accuracy and compressions are for networks pruned using descriptive pruning.

Pruning on position, we achieve an accuracy of 72.3% with a compression of 94%. In this case, we removed all actors except for the chairs of the company boards. Pruning on tenure, we achieve an accuracy of 70.29% with a compression of 95%, and pruning on age, we achieve an accuracy of 69.2% with a compression of 99%. In this case, we kept only the older executives. These accuracies are all better than the baseline prediction accuracy of 61% achieved by simply choosing the most common exchange.

For predicting the ECN sector, shown in Figure 3(b), the full network achieves accuracy of 40.4%. Here pruning based on both descriptive and structural properties perform well. When pruning based on age, we achieve accuracy of 40.2% with compression of 34%. In this case we kept the younger executives rather than the older ones. When pruning based on structure, we achieve accuracy of 39.7% and compression of 97% by keeping only the brokers. Figure 3(c) and (d) show similar results for the pruned APN networks, with many of the pruned networks achieving significantly higher accuracies than classifiers built from the full network. For both APN attributes, the network pruned on structure that achieved the best accuracy-compression tradeoff was the one that kept only the actors that were both hubs and brokers.

For both data sets, pruning on descriptive attributes and structure properties outperformed random pruning. One question this raised was whether or not the different pruning techniques were removing the same nodes and edges or different ones? To address the first question, Figure 4 shows the percentage of structural actor types (hubs, brokers (BRK), hubs and brokers (HBR), and other) preserved under various descriptive pruning strategies. These graphs show that for both data sets, the networks created using descriptive pruning contain a different mix of actors than those created using structural pruning. This supports our claim that structural pruning and descriptive pruning are two distinct methods for compressing networks and maintaining information rich nodes for prediction in affiliation networks.

6 Related Work

A large portion of the work in mining social networks has focused on analyzing structural properties of the networks. For a recent survey, see Newman [13]. 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., [5, 9, 3]). These papers attempt to identify the most influential nodes in the network. Domingos and Richardson [5] use a global, probabilistic model that employs the joint distribution of the behavior over all the nodes. Kempe et al. [9] 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. Liben-Nowell and Kleinberg [12] attempt to predict future interactions between actors using the network topology. In addition, Palmer et al. [14] propose an efficient method for approximating the connectivity properties of a graph.

Other work uses structural properties for both classification and clustering. Agrawal et al. [1] use the link structure of newsgroup social networks to classify user behavior within a newsgroup, specifically they identify whether a respondent agrees with a posting. Schwartz and Wood [16] create an email graph with edges corresponding to sets of shared interests and present an algorithm that analyzes the graph structure to cluster users with similar interests. Their approach derives a specialization subgraph from the relationship clusters.

Graph sampling and compression is also a relevant, active area of study. As we saw in section 5, random sampling did not generally lead to good prediction results. This finding agrees with that of Airoldi and Carley [2]. They find that pure network topologies are sensitive to random sampling. As mentioned earlier, graphs have been compressed using different local network measures [4]. A similar approach is to use frequently occurring subgraphs as proposed in [10].

There is also a related line of work which makes use of the descriptive attributes of the entities in the network for collective classification (e.g., [8, 18, 7]). While potentially applicable here as well, our focus is not on collective classification.

7 Conclusions

Exploring descriptive and structural pruning techniques together is needed for compact and accurate compression of networks. In this paper we showed how to use structural properties and descriptive attributes to prune social networks. We began by introducing a general framework for representing affiliation networks using relational algebra to formally express different network representations. We then used relational algebra expressions to define pruning strategies based on structural properties and descriptive attributes. Finally, we demonstrated the effectiveness of these pruning approaches on two real world data sets. While the networks resulting from structural pruning and descriptive pruning are quite distinct, both are viable approaches for reducing the size of a social network while still maintaining predictive accuracy on a set of target event attributes. Both approaches perform better than random sampling and lead to understandable, compressed networks that maintain (and in some cases increase) predict power.

References

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

[2] E. M. Airoldi and K. M. Carley. Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings. *SIGKDD Explorations Newsletter*, 7(2):13–22, 2005.

[3] M. Boguna and R. Pastor-Satorras. Epidemic spreading in correlated complex networks. *Physical review*, E 66(4), 2002.

[4] N. Deo and B. Litow. A structural approach to graph compression, 1998.

[5] P. Domingos and M. Richadson. Mining the network value of customers. In *ACM Intl. Conf. on Knowledge Discovery and Data Mining*, 2001.

[6] H. Garcia-Molina, J. Ullman, and J. Widom. *Database Systems*. Prentice Hall, New Jersey, 2002.

[7] L. Getoor. Link-based classification. In S. Bandyopadhyay, U. Maulik, L. Holder, and D. Cook, editors, *Advanced Methods for Knowledge Discovery from Complex Data*. Springer, 2005.

[8] D. Jensen and J. Neville. Data mining in social networks. In *National Academy of Sciences Symposium on Dynamic Social Network Analysis*, 2002.

[9] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *ACM Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

[10] N. S. Ketkar, L. B. Holder, and D. J. Cook. Subdue: compression-based frequent pattern discovery in graph data. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining*, pages 71–76, New York, NY, USA, 2005. ACM Press.

[11] A. J. Knobbe, M. de Haas, and A. Siebes. Propositionalisation and aggregates. In *Eur. Conf. on Principles of Data Mining and Knowledge Discovery*. Springer-Verlag, 2001.

[12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Intl. Conf. on Information and Knowledge Management*, 2003.

[13] M. Newman. The structure and function of complex networks. *IAM Review*, 45(2):167–256, 2003.

[14] 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.

[15] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

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

[17] L. Singh, L. Getoor, and L. Licamele. Pruning social networks using structural properties and descriptive attributes. In *IEEE International Conference on Data Mining*, pages 773–776, Washington, DC, USA, 2005. IEEE Computer Society.

[18] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Intl. Joint Conf. on AI*, 2001.

[19] S. Wasserman and K. Faust. *Social network analysis: methods and applications*. Cambridge University Press, Cambridge, 1994.

Peer-to-Peer Information Search: Semantic, Social, or Spiritual?*

Matthias Bender, Tom Crecelius, Mouna Kacimi,
Sebastian Michel, Josiane Xavier Parreira, Gerhard Weikum

Max-Planck Institute for Informatics
Saarbruecken, Germany
{mbender, tcrecel, mkacimi, smichel, jparreir, weikum}@mpi-inf.mpg.de

Abstract

We consider the network structure and query processing capabilities of social communities like bookmarks and photo sharing communities such as del.icio.us or flickr. A common feature of all these networks is that the content is generated by the users and that users create social links with other users. The evolving network naturally resembles a peer-to-peer system, where the peers correspond to users. We consider the problem of query routing in such a peer-to-peer setting where peers are collaborating to form a distributed search engine. We have identified three query routing paradigms: semantic routing based on query-to-content similarities, social routing based on friendship links within the community, and spiritual routing based on user-to-user similarities such as shared interests or similar behavior. We discuss how these techniques can be integrated into an existing peer-to-peer search engine and present a performance study on search-result quality using real-world data obtained from the social bookmark community del.icio.us.

1 Introduction

Peer-to-peer (P2P) information management and search is intriguing for scalability and availability. In addition, a P2P network would be a natural habitat for exploiting the "social wisdom" of its users. We envision a P2P system where each user runs a peer computer (e.g., on her PC, notebook, or even cell phone) and shares information within a large community. Each peer would be a full-fledged data management system for the user's personal information, scholarly work, or data that the user may harvest (and cache) from Internet sources such as news, blogs, or specialized Web portals. Each peer would also have a local search engine, which could be very powerful (e.g., using advanced NLP, machine learning, and ontologies), given that it operates on the user's relatively small-sized information collection on a dedicated computer, and could be highly customized to the user's individual interests and behavior. The Minerva platform developed in our group [4] follows this paradigm; other projects along the same lines include, for example, pSearch [37], Alvis [23], and BestPeers [18].

As a futuristic application scenario consider millions of users who use their mobile devices to record photos and videos of all kinds of real-world events ranging from business meetings to vacation trips. Such digital-perception information can be easily annotated with speech and device-generated metadata such as GPS and time

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

*This work is partially supported by the EU project SAPIR.

coordinates. Moreover, all this data could be made accessible on a P2P network instantaneously, for search and further annotation - so-called "social tagging" - by other users. For example, thousands of tourists on the Forum Romanum can immediately share their photos and annotations, so that an uninitiated tourist could immediately receive explanations about some lesser known remains from the annotations of other, more knowledgeable users. The P2P search and social-networking technology that underlies such a scenario would be embedded in the application software and be virtually invisible to the end-users.

In such P2P settings, queries would first be executed locally, on the peer where the query is issued. This would utilize the locally available information and powerful, personalized search capabilities. In some settings, this may be a local cache of annotated photos or MP3 files; in others, it could be a collection of personally relevant Web pages that have been compiled by thematically focused crawling and subscriptions to feeds. If the local search does not return satisfactory results, the peer should consider forwarding the query to a small number of judiciously chosen other peers. This step towards collaborative search is known as the *query routing* decision. It should consider both the expected benefits of obtaining better information from other peers and the communication and execution costs of involving these peers. The literature on P2P information retrieval and other forms of distributed IR contains many proposals for query routing strategies; see, e.g., [25, 14, 16, 22, 30, 5, 26, 3].

The routing decision is usually driven by various forms of precomputed (and incrementally maintained) routing indices, peer-content synopses, or distributed directories, which in turn can influence the topology of the P2P overlay network leading to so-called semantic overlay networks (SONs) [11, 31, 2, 21, 12, 1].

In the current paper, we do not make any assumptions about this infrastructure or the overlay topology, and rather assume that the query routing decision has all the information about other peers that it needs and chooses peers solely by benefit/cost considerations. We will disregard the cost aspects for this paper and focus on the much less explored benefit issues.

We investigate three broad families of strategies:

- *Semantic query routing*: The peers to which a query is forwarded are chosen based on the *content similarity* between the query and the data held by the candidate target peers (or the corresponding peer synopses).

- *Social query routing*: The target peers are chosen based on *social relationships* like the explicitly listed friends of the query initiator or peers that belong to the same explicit groups.

- *Spiritual query routing*: The target peers are chosen based on *behavioral affinity* such as high overlap in tag usage, bookmarked pages, or commenting and rating activity. This aims to capture "brothers in spirit", hence the name.

We refer to the first family as "semantic" as the content comparison could take into account metadata (e.g., schema mappings), ontology-based similarities, and other aspects that go beyond purely syntactic or statistical measures. For simplicity, the current paper considers only keyword queries (referring to text terms or user-provided tags) and consequently uses simple measures of (IR-style) statistical similarity, but the approach could be enriched and generalized. The second and the third approach are closely related and could be easily confused. We refer to "social search" when explicit friendship or other social-networking relations are used, and we refer to "spiritual search" when considering users' tagging, bookmarking, rating, and other behaviors.

This paper discusses how these three approaches can be used in P2P query routing, and how effective they are for delivering high-quality results. As we consider keyword queries, we will use IR quality measures like precision and recall. We also present hybrid strategies that combine elements from both semantic and social or semantic and spiritual search. The rest of the paper is organized as follows. Section 2 briefly reviews the state of the art on P2P information search and its relation to social networks. Section 3 presents the Minerva system architecture, which is our testbed and serves as a representative of the general architectures to which our work applies. Section 4 introduces our query routing strategies in more detail. Section 5 presents an experimental comparison of different strategies, using data extracted from the popular social-tagging site *del.icio.us*. Section 6 points out lessons learned and future work.

2 Related Work

One of the fundamental functionalities that a P2P information system must provide is to identify the most "appropriate" peers for a particular query, i.e., those peers that are expected to locally hold high-quality results for the query. This task is commonly referred to as query routing, sometimes also as resource or collection selection. We stress that query routing is more challenging than it may appear at first sight: the set of peers to be contacted is not simply the set of all peers that store relevant index data. Such a set could contain a very large number of peers and contacting all of them would be prohibitive. While there exist a number of approaches for query routing in the literature on distributed IR — e.g., CORI [9], GlOSS [16], and methods based on statistical language models [34] — these were typically designed for a stable and rather small set of collections (e.g., in the context of metasearch engines). These techniques usually assume that the document collections are disjoint, which is a rather unrealistic assumption in P2P systems where the peers are compiling their content (e.g., by crawling the Web) at their discretion. In [5, 27] we have proposed the usage of overlap aware query routing strategies. The proposed methods use compact data synopses such as Bloom filters or hash sketches to estimate the mutual overlap between peers to avoid querying peers that provide basically the same information, which would waste both processing power and network resources.

The statistical summaries describing a peer are usually organized on a per-term basis, indicating the expected result quality of a peer's collection for a given term. This limitation is considered unavoidable, as statistics on all term pairs would incur a quadratic explosion, leading to a breach with the goal of scalability. On the other hand, completely disregarding correlations among terms is a major impediment: for example, consider the following extreme scenario. Assume peer p_1 contains a large number of data items for each of the two terms a and b separately, but none that contains both a and b together. Judging only by per-term statistics, state-of-the-art query routing approaches would reach the conclusion that p_1 is a good candidate peer for the query $\{a, b\}$, whereas the actual result set would be empty. In [26, 6], we present a routing method that uses multi-key statistics to improve the query routing performance. We propose the usage of a distributed query-log analysis to discover frequently co-occurring keys (terms) that are candidates for being considered as additional keys in the distributed directory. To decrease the directory load, we introduce a pruning technique to avoid considering unnecessary key-sets.

Social networks have recently emerged in P2P systems to address several issues such as improving content discovery [13, 8, 19], reducing latency and speeding up downloads [32, 38, 35], and designing trust models [24, 17]. In the following, we briefly present some approaches towards P2P search.

Pouwelse et al. [32] propose Tribler, a social-based P2P overlay on top of BitTorrent. It connects peers based on their similar "tastes" instead of considering similar files. Thus, peers exploit their social links and invoke the help of their friends to improve content discovery and download cost. Similarly, Fast et al. [13] propose using user interests to build social groups in a P2P network. Users sharing the same type of files are connected to each other even though their contents do not overlap. The main goal of this approach is to capture important aspects of download behavior by connecting peers to the potential providers of their required files.

Other social P2P networks are based on peer request traces. A peer uses request relationships to other peers to construct social links to them. Sripanidkulchai et al. [35] implement a performance enhancement layer on top of the flooding-based content location mechanism of Gnutella. Each peer creates and maintains its shortcuts list based on its request trace. Shortcuts are ranked according to some metrics such as the probability of providing relevant content, latency of the path to the shortcut, available path bandwidth, shortcut load, etc. The work presented by Tempich et al. [38] considers query traces to create a human social network. It defines a query routing strategy in which peers observe which queries are successfully answered by other peers and remember

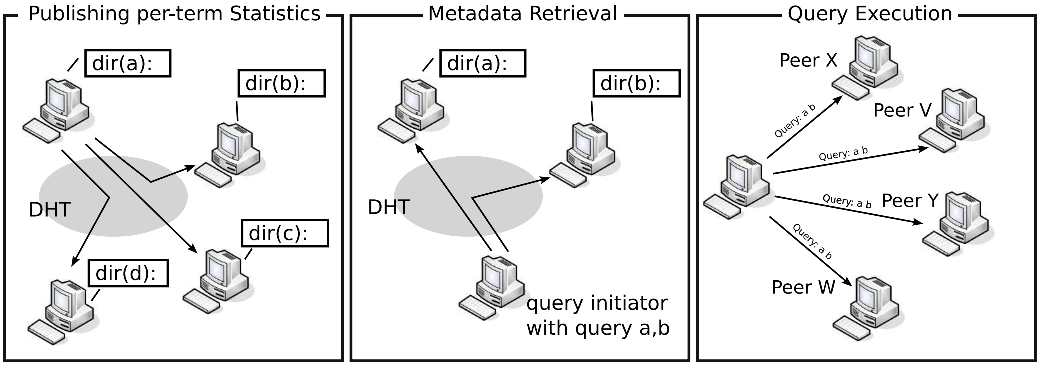


Figure 1: Metadata Dissemination, Query Routing, and Query Execution in Minerva.

these peers in future query routing decisions.

Borch et al. [8] present a social P2P search infrastructure which groups peers based on the similarity of their keyword searches. The authors describe different application scenarios including distributed bookmark sharing in which users create bookmarks, describe them using tags, and share them with friends and colleagues. The basic idea is to send queries to peers likely to have interesting resources. Khambatti et al. [19] introduce the notion of peer communities that consist of active peers involved in sharing, communicating, and promoting common interests. These communities are self-organizing using distributed formation and discovery algorithms.

3 Minerva

We have developed a P2P Web search engine coined Minerva, released as open source and available under *http://www.minerva-project.org*. We envision a network of peers, each with a local index and a local query processor, that are crawling the Web independently, for example, to harvest blogs or scientific publications according to the user's thematic profile. Minerva maintains a metadata directory that is layered on top of a distributed hash table (DHT) [36, 33]. It holds very compact, aggregated summaries of the peers' local indexes and only to the extent that the individual peers are willing to disclose. A query initiator selects a few most promising peers based on their published per-term summaries, e.g., by executing a distributed top-k algorithm like [10, 28]. Subsequently, it forwards the complete query to the selected peers which execute the query locally. This query execution does not involve a distributed top-k query execution since each peer maintains a full-fledged local index with all information necessary to execute the query locally. Finally, the results from the various peers are combined at the querying peer into a single result list.

Figure 1 illustrates the Minerva approach. First, every peer publishes per-term summaries (*Posts*) of its local index to the directory. The DHT (and its replication mechanism) determines the peer(s) currently responsible for this term. This peer (or these peers in the case of replication) maintains a *PeerList* of all postings for this term from across the network. Posts contain contact information about the peer who posted a summary together with statistics to calculate IR-style measures for a term (e.g., the size of the inverted list for the term, the average score for the term's inverted list entries, or other statistical measures). These statistics are used to support the query routing decision, i.e., determining the most promising peers for a query.

Minerva facilitates easy integration of new query routing strategies, like the ones proposed in this paper. For instance, users' bookmarks can be crawled and indexed, and their terms can then be posted to the distributed metadata directory. Similarly, tags used to describe the bookmarks can be stored in the directory. This supports semantic query routing. For the social and spiritual query routing, the Minerva framework can be extended by keeping, at each peer, a list of peers that are related either by social relationship or behavioral affinity. Note that these lists tend to be very small, relative to the size of the network; so the approach scales up well.

In the spirit of social tagging communities, users can manually add arbitrary attribute-value annotations by a single mouse click. For example, users might rate Web pages or blogs with annotations such as *rating=5*. Additional annotations may be automatically generated from the content, such as *author=weikum* or *conference=ICDE*. These annotations are also indexed and become part of the directory; so users can explicitly query for documents with *rating=5* and also combine such conditions with query keywords.

4 Query Routing in Social P2P Networks

4.1 Semantic Query Routing

The peers to which a query is forwarded are chosen based on the *content similarity* between the query and the data held by the candidate target peers (or the corresponding peer synopses). The query is represented by its keywords – *terms* in IR jargon –; the data of a peer can be represented by its terms or its tags or a combination of both. With each term and each tag we can also associate some precomputed frequency statistics, e.g., how often a term or tag has been used by a given peer and how often it is used in the overall P2P network. Following query-routing terminology, we refer to the total frequency of tag or term t at peer p_j as the *document frequency* $df_j(t)$; this is the number of bookmarked pages in p_j's collection that contain or are tagged with term/tag t.

Semantic query routing estimates the benefit for different candidates based on the sum of document frequencies for the query terms (as determined by the best entries from the term- or tag-specific directory entries fetched via DHT lookups), and chooses the highest-ranked peers according to this measure. Alternatively, one could also employ more sophisticated methods such as CORI [9] that uses, in addition to the document frequency, several dampening and smoothing techniques partially based on the notion of collection frequencies, i.e., the number of peers that have bookmarked pages that contains a particular term.

4.2 Social Query Routing

The target peers are chosen based on *social relationships*. We assume that there is an explicit *friends* relation among peers, and we choose target peers for forwarding a query issued at peer p_j to be the "best" friends of p_j, provided the degree of friendships are quantified (e.g., based on the frequency of interactions between peers in the recent past). If there is no quantitative measure for friendship strength, then we simply choose a random subset of friends when we want to limit the number of target peers, or all friends when there is no limit.

4.3 Spiritual Query Routing

The target peers are chosen based on *behavioral affinity* such as high overlap in tag usage, bookmarked pages [7], or commenting and rating activity. We could use an information-theoretic measure, the Kullback-Leibler divergence (relative entropy) [20], on the tag frequency distributions of a peer's bookmarked pages (possibly combined with rating information), and would quantify the similarity for each pair of peers. We can then use such a similarity measure to cluster peers that are spiritually close to each other. A simpler approach with the same intention considers the overlap in the bookmarked pages among peers. This can be efficiently computed in a P2P environment using distributed algorithms on compact synopses like Bloom filters [5, 27]. Spiritual query routing for a query initiated at peer p_i then chooses the peers p_j with the highest estimated $overlap(p_i, p_j)$.

4.4 Hybrid Strategies

All the aforementioned routing strategies can be combined into hybrid methods. Here we outline only some straightforward approaches and leave more sophisticated combinations for future work. The goal of peer selection is to identify the top-k peers for a particular query. A hybrid approach would select k_i peers with strategy

S_i so that $\sum_i k_i = k$. The choice of the single k_i values is a nontrivial problem (cf. [29]). A simple approach would, for example, use $k_1 = k_2, ...$ and a round-robin selection.

Combining the social routing strategy with a spiritual routing strategy would, for instance, decrease the risk of obtaining mediocre results when the query does not fit with the friends' thematic interests in a purely social routing strategy.

4.5 Orthogonal Issues

Besides the aforementioned query routing concepts that aim to find promising peers for a particular information need, an overlap-aware technique [5, 27] can be employed to eliminate redundancy in the query evaluation. For instance, it does not make sense to query both peers A and B if it is known that both have (almost) the same information or A's collection is a subset of B's collection.

5 Experiments

5.1 Data Collection

We have crawled parts of del.icio.us[1] with a total of $13,515$ users, $4,582,773$ bookmarks, and $152,306$ friendship connections. In addition, we have actually crawled and indexed the actual HTML pages where the bookmarks point to, giving us the possibility to execute both term-based and tag-based queries.

Each peer in our experiments corresponds to exactly one user. The local collection of a peer consists of the bookmarked pages, including their actual contents, and the user-provided tags for each page.

5.2 Queries

For the workload we needed realistic queries and their association with specific users. Query logs with this kind of information are not publicly available. Therefore, we generated queries based on the users' tags in a way that the queries reflect the user interests. For a particular user we consider those tags that frequently co-occur for the same bookmarks.

More precisely, to generate the benchmark queries, we first identified the top ξ users in terms of bookmark-set cardinalities. Then we considered those tag pairs that were used together at least ζ times and not more than ψ times by the selected users. The first constraint is needed to eliminate rare tag pairs. The second constraint is used to eliminate tag pairs that have a stopword character. For our experiments we chose $\xi = 5$, $\zeta = 200$, and $\psi = 900$. Using this technique we identified 24 queries with two tags, such as "music media", "web design", "mac apple", and "tech reference".

5.3 Quality Measures

There are no standard queries and no relevance assessments available for the pages bookmarked in del.icio.us. We consider two different approaches for defining some notion of "ground truth": a hypothesized ideal search result to which our strategies can be compared.

(i) As a first approach, we use pages bookmarked by the query initiator as ground truth. Consider a multi-keyword query $Q = q_1, q_2, ..., q_m$. The query initiator retrieves the top-k pages from each of the peers selected during the query routing phase. Then, to estimate the quality of the retrieved pages, the initiator compares the obtained results with the pages she has bookmarked and tagged with tags $q_1, q_2, ..., q_m$. The rationale behind this evaluation is that the fact that a user has bookmarked a page can be interpreted as relevance judgment.

[1] *http://del.icio.us*

(ii) As an alternative approach, which is independent of the query initiator, we consider all pages that are bookmarked in the system and tagged (by some user) with all the query keywords as relevant. The goal for the query execution then is to maximize the number of results from this pool of relevant pages.

For the first approach, the "relevance judgments" highly depend on the query initiator. Thus, we have to select as query initiators "power users" with a sufficiently large number of bookmarks. We first select a query by choosing a frequent tag pair. Then we rank peers that have at least 50 friends according to the number of bookmarks that are tagged with the chosen pair. For each query (i.e., keyword pair) we consider the top-5 peers as query initiators, i.e., we execute the same query five times to remove the influence of an accidentally bad choice for one of the initiators.

The second approach allows for relevance assessment that is independent of the query initiator, whereas the first approach depends on the choice of the query initiator. However, the social and the spiritual routing strategies depend on the query initiator anyway, as, for instance, executing a query related to pop music on a peer that is primarily interested in soccer would not return good results by design.

Once a peer receives an incoming query request, it executes the query locally and returns *all* bookmarked pages that are *tagged* with the keywords in the query. In a real-world system one would try to return only the top-k results by some meaningful ranking. However, as we deal with personalized search here, it is not straightforward to apply a standard scoring model. Therefore, we let peers return all bookmarked pages that match the query.

The same situation occurs when we merge the result lists returned by the queried peers: as there is no widely agreed merging strategy, we assess the quality of the union of the returned results.

5.4 Strategies under Comparison

For multi-keyword queries of the form $Q = \{t_1, .., t_m\}$ we evaluate the retrieval quality, measured by recall (relative to the ground truth explained in the previous subsection), of the following strategies:

- **Semantic Routing based on Tags:** We rank peers according to the sum of document frequencies, i.e., the score of a peer p_i is given by $\sum_{t \in Q} df_i(t)$ where $df_i(t)$ is the number of bookmarks in peer p_i's collection that are tagged with t, cf. Section 4.1.

- **Semantic Routing based on Terms:** We rank peers according to the sum of document frequencies, similar to the tag based semantic routing, but here we consider terms instead of tags.

- **Social Routing:** We let the query initiator send the query to the top friends where the friends are ranked according to the number of bookmarks they have.

- **Spiritual Routing:** For spiritual closeness we consider the overlap in the bookmarks.

- **Hybrid between Semantic and Spiritual Routing:** This hybrid strategy combines the routing results (peer rankings) obtained from the semantic and spiritual routing strategies in a round-robin manner, ignoring duplicates.

- **Hybrid between Semantic and Social Routing:** This is a combination of the semantic and the social routing results using a round-robin selection process, ignoring duplicates.

- **Hybrid between Spiritual and Social Routing:** This is a combination of the spiritual and the social routing results using a round-robin selection process, ignoring duplicates.

5.5 Experimental Results

Figure 2 shows the average recall for the benchmark with 120 queries (24 distinct queries, each issued by 5 different peers) when considering the query initiator's bookmarks as the ground truth. The semantic routing strategy is the clear winner. The spiritual routing strategy performs reasonably well but cannot reach the performance of the semantic routing strategy. For instance, when asking 10 peers, the semantic routing strategy achieves a recall of nearly 16% whereas the spiritual strategy achieves approximately 8% recall. The social routing strategy performs worse than all other strategies. Surprisingly, the term-based semantic routing strategy performs poorly. This is probably due to the particular nature of the queries that have been created based on the most popular tags as many tags are not "appropriate" search terms. Examples are "Task Organizing" tags [15] like "toread" or "jobsearch". [15] gives a nice overview on the different functions that tags can have.

The relative order of the hybrid strategies follows that of the pure strategies: the semantic-spiritual strategy is the best hybrid strategy, followed by the semantic-social strategy, and the spiritual-social strategy performs worst but still better than the purely social strategy.

Figure 3 shows similar results for the second choice of ground truth with bookmarked pages that are tagged with the query words as relevant. The results confirmed our findings from the first experiment; so no further discussion is needed here.

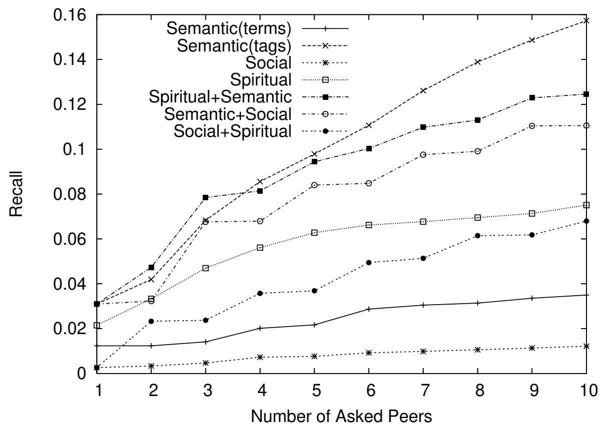


Figure 2: Average Recall: considering the query initiator's bookmarks that match the query tags as relevant.

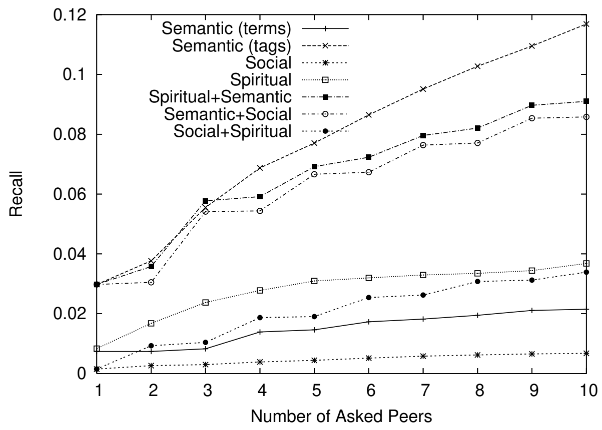


Figure 3: Average Recall: considering all bookmarks that match the query tags as relevant.

6 Lessons Learned and Future Work

Our experiments have shown that the semantic routing strategies that use per-tag peer summaries are superior to all other strategies. The social routing strategy performed very poorly in our experiments, and it is disappointing to see that it provided hardly any relevant results.

To understand this poor performance we have analyzed the content overlap among peers that are related by friendship connections. For each user, we have calculated the overlap between her bookmarks and the bookmarks from her friends. It turned out that the overlap is surprisingly small: considering only users that have at least one friend, the mean value is about 7%, i.e., half of the peers share less than 7% of their bookmarks with their neighbors. The minimum overlap observed was 0.03 %, the first and third quartiles were 2.8% and 14.5%, respectively. Theses low numbers partially explain the bad performance of the social routing strategy. In our experiments, for half of the users, a recall of at most 7% would be obtained if we had asked all their friends. Since we limited the number of friends queried to 10, the obtained recall was even lower.

We believe that this phenomenon is due to the particular usage of the friend relationships in del.icio.us. It seems that users establish a new friendship connection when the bookmarks tagged by the new friend are considered as interesting, and then the user does not care anymore about tagging the same pages. This interesting feature of such networks may need further exploration.

Note that the social routing strategy does not require any global information like the semantic strategy and the spiritual strategy. The semantic strategy needs a global mapping from tags (or terms) to per-peer summaries that cause some maintenance cost (to update the DHT-based directory). The spiritual routing strategy requires continuous peer meetings to learn about thematically close peers, although these information exchanges could probably be piggybacked on messages that are sent anyway on behalf of user queries.

Our intention in this paper was to outline our framework for semantic, social, and spiritual query routing, identify technical issues, and shed some light into the experimental behavior of these P2P routing strategies within social networks. Our findings clearly dampen the optimism about social networks being able to boost search result quality in a P2P network. More traditional content-oriented strategies were found to be way superior. However, our observations and insights are clearly preliminary at this point, and should stimulate further research in this area.

References

[1] K. Aberer and P. Cudré-Mauroux. Semantic overlay networks. In *VLDB*, page 1367, 2005.

[2] K. Aberer, P. Cudré-Mauroux, M. Hauswirth, and T. V. Pelt. Gridvine: Building internet-scale semantic overlay networks. In *International Semantic Web Conference*, pages 107–121, 2004.

[3] R. Baeza-Yates, D. Puppin, and R. Perego. Incremental caching for collection selection architectures. In *Infoscale*, 2007.

[4] M. Bender, S. Michel, J. X. Parreira, and T. Crecelius. P2p web search: Make it light, make it fly (demo). In *CIDR*, pages 164–168, 2007.

[5] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in p2p search engines. In *SIGIR*, pages 67–74, 2005.

[6] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. P2P content search: Give the web back to the people. In *5th International Workshop on Peer-to-Peer Systems (IPTPS 2006)*, 2006.

[7] M. Bender, S. Michel, G. Weikum, and C. Zimmer. Bookmark-driven query routing in peer-to-peer web search. In *Workshop on Peer-to-Peer Information Retrieval*, 2004.

[8] N. Borch. Social peer-to-peer for social people. In *The International Conference on Internet Technologies and Applications*, 2005.

[9] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *SIGIR*, pages 21–28, 1995.

[10] P. Cao and Z. Wang. Efficient top-k query calculation in distributed networks. In *PODC*, pages 206–215, 2004.

[11] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. In *AP2PC*, pages 1–13, 2004.

[12] C. Doulkeridis, K. Nørvåg, and M. Vazirgiannis. The sowes approach to p2p web search using semantic overlays. In *WWW*, pages 1027–1028, 2006.

[13] A. Fast, D. Jensen, and B. N. Levine. Creating social networks to improve peer-to-peer networking. In *KDD*, pages 568–573, 2005.

[14] N. Fuhr. A decision-theoretic approach to database selection in networked ir. *ACM Trans. Inf. Syst.*, 17(3):229–249, 1999.

[15] S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[16] L. Gravano, H. Garcia-Molina, and A. Tomasic. Gloss: Text-source discovery over the internet. *ACM Trans. Database Syst.*, 24(2):229–264, 1999.

[17] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC*, pages 411–426, 2006.

[18] P. Kalnis, W. S. Ng, B. C. Ooi, and K.-L. Tan. Answering similarity queries in peer-to-peer networks. *Inf. Syst.*, 31(1):57–72, 2006.

[19] M. Khambatti, K. D. Ryu, and P. Dasgupta. Structuring peer-to-peer networks using interest-based communities. In *DBISP2P*, pages 48–63, 2003.

[20] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[21] A. Löser, C. Tempich, B. Quilitz, W.-T. Balke, S. Staab, and W. Nejdl. Searching dynamic communities with personal indexes. In *International Semantic Web Conference*, pages 491–505, 2005.

[22] J. Lu and J. P. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *CIKM*, pages 199–206, 2003.

[23] T. Luu, F. Klemm, I. Podnar, M. Rajman, and K. Aberer. Alvis peers: a scalable full-text peer-to-peer retrieval engine. In *P2PIR '06: Proceedings of the international workshop on Information retrieval in peer-to-peer networks*, pages 41–48, New York, NY, USA, 2006. ACM Press.

[24] S. Marti, P. Ganesan, and H. Garcia-Molina. DHT routing using social links. In *IPTPS*, pages 100–111, 2004.

[25] W. Meng, C. T. Yu, and K.-L. Liu. Building efficient and effective metasearch engines. *ACM Comput. Surv.*, 34(1):48–89, 2002.

[26] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, and C. Zimmer. Discovering and exploiting keyword and attribute-value co-occurrences to improve p2p routing indices. In *CIKM*, pages 172–181, 2006.

[27] S. Michel, M. Bender, P. Triantafillou, and G. Weikum. Iqn routing: Integrating quality and novelty in p2p querying and ranking. In *EDBT*, pages 149–166, 2006.

[28] S. Michel, P. Triantafillou, and G. Weikum. Klee: A framework for distributed top-k query algorithms. In *VLDB*, pages 637–648, 2005.

[29] H. Nottelmann and N. Fuhr. Combining cori and the decision-theoretic approach for advanced resource selection. In *ECIR*, pages 138–153, 2004.

[30] H. Nottelmann and N. Fuhr. Comparing different architectures for query routing in peer-to-peer networks. In *ECIR*, pages 253–264, 2006.

[31] J. X. Parreira, S. Michel, and G. Weikum. p2pdating: Real life inspired semantic overlay networks for web search. *Inf. Process. Manage.*, 43(3):643–664, 2007.

[32] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 2007.

[33] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware*, pages 329–350, 2001.

[34] L. Si, R. Jin, J. P. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In *CIKM*, pages 391–397, 2002.

[35] K. Sripanidkulchai, B. M. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, 2003.

[36] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, pages 149–160, 2001.

[37] C. Tang, Z. Xu, and M. Mahalingam. psearch: information retrieval in structured overlays. *Computer Communication Review*, 33(1):89–94, 2003.

[38] C. Tempich, S. Staab, and A. Wranik. Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In *WWW*, pages 640–649, 2004.

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398