Groups Without Tears: Mining Social Topologies from Email

Diana MacLean Sudheendra Hangal Seng Keat Teh
Monica S. Lam Jeffrey Heer
Computer Science Department
Stanford University
{malcdi, hangal, skteh, lam, jheer}@cs.stanford.edu

ABSTRACT

As people accumulate hundreds of "friends" in social media, a flat list of connections becomes unmanageable. Interfaces agnostic to social structure hinder the nuanced sharing of personal data such as photos, status updates, news feeds, and comments. To address this problem, we propose *social topologies*, a set of potentially overlapping and nested social groups, that represent the structure and content of a person's social network as a first-class object. We contribute an algorithm for creating social topologies by mining communication history and identifying likely groups based on co-occurrence patterns. We use our algorithm to populate a browser interface that supports creation and editing of social groups via direct manipulation. A user study confirms that our approach models subjects' social topologies well, and that our interface enables intuitive browsing and management of a personal social landscape.

Author Keywords

Social topology, social graph, data sharing, access control

ACM Classification Keywords

H.5.3 Group and Organization Interfaces

General Terms

Design, Human Factors, Management

INTRODUCTION

Today's online experience depends increasingly on social interaction. Sites such as Facebook and LinkedIn have evolved from recreational to socially essential; media-sharing platforms such as Flickr and LastFM attract large numbers of users; collaborative productivity tools such as Google Docs continue to grow in popularity. However, while our online data and behavioral patterns are increasingly contextualized by our personal social networks, we lack a corresponding mechanism for defining, organizing, and maintaining different parts of these networks.

Current social networking activity is characterized by coarse-grained information sharing, leading to frequent under-sharing and over-sharing. To target information to specific segments of their social networks, users must engage in time-consuming and potentially redundant tasks, such as constructing contact groups on Gmail, maintaining friends' lists on Facebook, or enumerating all people with whom to share a Flickr album. Most users find these tasks tedious and painful to do [19]. Moreover, the creation of static social groups fails to capture the nuances of social context that exist in real life. As human relationships evolve and the strength of social ties change, the memberships of these social groups must also be dynamically updated.

In response, we introduce the concept of a *social topology*: the structure and content of a person's social affiliations, consisting of a set of possibly overlapping and nested social groups. Permitting *overlapping* groups allows a social topology to accurately represent people who perform multiple roles simultaneously in one's life, such as the colleague who is also a hiking buddy, or a family member who shares a particular hobby. Permitting *nested* groups allows representation of social affiliations at various levels of granularity, such as best friends within friends. Although a person may have an implicit mental model of her social topology, its nuanced complexity and sheer volume make it difficult to capture and maintain manually. We hypothesize that an algorithm for automatic social topology extraction, coupled with an interface for storage and maintenance, can facilitate social organization and information sharing.

We observe that many people already have a large dataset that *latently* and *accurately* reflects their social topology over time: personal email.

Email has been a common medium for online social interaction for many years. It is estimated that there are over 1.3 billion email users worldwide and that this number will grow to 1.8 billion by 2012 [17]. Mainstream users routinely acquire and store large volumes of email, thanks to the availability of cheap storage and the ubiquity of providers offering free email service.

This research is supported in part by the NSF POMI (Programmable Open Mobile Internet) 2020 Expedition Grant 0832820, a Stanford Graduate Fellowship, the Stanford Clean Slate Program, and Deutsche Telekom.

IUI 2011, February 13–16, 2011, Palo Alto, California, USA.

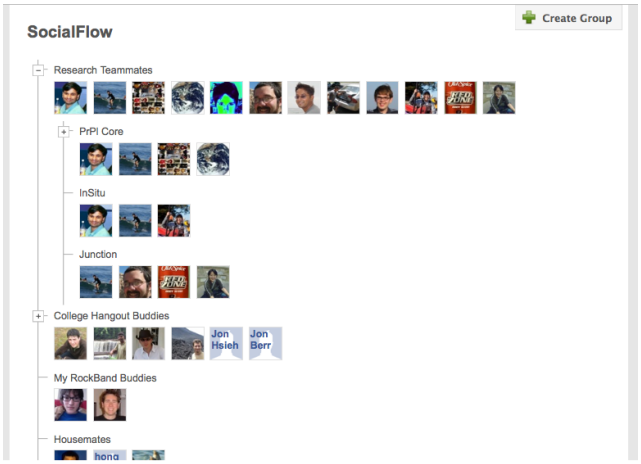

Figure 1. Our social topology browser, showing social groups automatically mined from e-mail data. Individuals can be members of multiple groups; subgroups within groups are nested hierarchically.

Email can be viewed as a social system in which users routinely express sharing rights over their information. Because each message defines a specific set of recipients, email captures, *in situ*, both social relationships and changes in social relationships over time at fine granularity. Thus, email offers a fine-grained sharing model closest to the one we imagine for social networks. While mailing lists are a convenient grouping mechanism for email, they are typically used for formal group membership. Many more groups, such as friend and family units, are formed in an ad-hoc manner.

Using this insight, we have built a system for constructing a user's social topology from their email, using a combination of data mining techniques and user input to finesse the topology's accuracy. Users may tune automatically generated topologies, and the system can semi-automatically maintain topologies over time by highlighting changes in social structure. We note that while we have used email as a primary and motivating dataset, our approach is easily extended to incorporate other forms of communications media.

Our research contributions include:

- The introduction of *social topologies* as first-class objects that represent social groups at multiple granularities.
- A new browser interface for social topologies (Figure 1).
- A novel and efficient algorithm for constructing a social topology from e-mail data. The algorithm detects community structure among a person's contacts, and is distinctive in its inference of overlapping and nested groups.
- A publicly available system called SocialFlows, accessible as a web application[1], that lets users interact with the social topology deduced from their email data. SocialFlows also allows users to export the created groups to Facebook as friends lists and groups, and to Gmail as contact lists.
- A user study evaluating the quality and accuracy of our system. Results suggest that our algorithm models users' social topologies sufficiently well, and that our interface enables intuitive topology browsing and management.

[1]Available at http://mobisocial.stanford.edu/socialflows

The rest of this paper proceeds as follows: we first survey related work, then discuss our motivation and techniques for extracting social topologies from email. We next present the details of our topology mining algorithm. Next, we describe our user interface for browsing and editing generated topologies, including a brief description of our web application implementation. Finally, we present the design and results of our user study.

RELATED WORK

To contextualize our research, we discuss prior work in social network analysis on community detection, and visualization of personal contacts.

Detecting Overlapping Communities in Social Networks

Social network analysis is a research area that has received significant attention, especially in the last several years. The correspondence between real-world social relations and social media ties has been confirmed on several occasions [6, 20]. The specific problem of community detection has been studied both for social graphs where a global view of the network is available [4, 10, 11, 12, 13, 15, 20, 26], and for egocentric networks, where only one individual's view is available [2, 6, 12].

Clustering-Based Approaches

A plethora of work exists on clustering algorithms for social networks, and we do not discuss them in depth here; for an overview, see Wasserman and Faust [24]. We note that standard methods of hierarchical agglomerative clustering are not adequate for our purposes, since they partition the network into non-overlapping communities. Therefore, we focus only on prior work that allows individuals to be classified into multiple communities. We further note that these methods, unlike ours, do not have a quantitative notion of subsumption, we use in our inference of nested groups.

Huberman et al. analyze a network of 485 people's email activity within an organization over four months [20]. Unlike our model, edges are drawn between the sender and each message recipient, but not between co-recipients. Eventually, edges between sender-recipient pairs beneath a threshold are dropped, and an algorithm based on edge betweenness centrality [24] detects non-overlapping groups. The algorithm runs multiple times, breaking ties randomly, and aggregates the results. Overlapping clusters are detected by weighting each individual's membership in a community by the frequency of that individual appearing in the cluster over all of the algorithm's iterations.

Palla et al. present an algorithm to identify overlapping communities in unweighted networks [16]. They consider globally oriented networks only, and do not consider email datasets. However, their algorithm bears a theoretical parallel to ours: potential communities are generated from smaller graph units (in their case, cliques; in ours, frequent subsets), resulting in hypothetical communities that may not exist in the origi-

nal graph. Their algorithm requires significant computation time, which would likely be unsuitable for widespread usage by consumers.

Non-Clustering-Based Approaches

Moody and White present a method of *cohesive block modelling* in which graph nodes are recursively clustered into "blocks" based upon their cohesiveness in the graph [13]. The resulting blocks tend to be nested, but overlapping blocks are possible.

Performing text analysis on social data (such as email) provides additional axes along which to group individuals. Culotta et al. take a unique approach in which a social graph constructed from an e-mail corpus is augmented with data mined from the Internet [2]. Links on the social graph are unweighted and extracted from e-mail headers, text, and online webpages. Each individual is tagged with keywords particular to his/her expertise. Given sufficient social context, one can imagine using such keywords to refine socially similar groups derived by our algorithm.

In later work, McCallum et al. focus on developing machine learning methods for discovering individual *roles* in a social network [12]. Given an e-mail corpus, their method models topic distributions over sender-recipient pairs. All sender-recipient pairs relevant to a topic can then be queried from the graph. McCallum et al. analyze both globally-oriented and egocentric networks with promising empirical results.

Zhou et al. propose generative Bayesian models to mine "semantic communities" within the Enron email corpus [26]. Resulting communities are labeled with a topic description.

Recently, Gmail offers an auto-suggest feature for email recipients. Once the user has typed in a few recipients for a message, Gmail suggests other candidates based on implicitly derived groups [18]. These implicit groups are not exposed to the user and cannot be directly viewed or edited. Our algorithm identifies *socially notable* groups in one's social topology, a function not provided by Gmail's algorithm.

Our Approach

To our knowledge no prior work focuses on identifying *overlapping* and *nested* communities in an egocentric network. In fact, our approach bears more similarity to prior work on association rule mining by Agrawal et al. [1] than to social network analysis. Moreover, leveraging communications data (such as email) yields a contextually rich social graph: incentives and costs involved in repeated communications inform the meaningfulness of ties more strongly than a one-time "friending".

Visualizations of Personal Contacts

Researchers have developed a number of visualizations of one's personal contacts. For example, Heer and Boyd's Vizster depicts egocentric networks extracted from online social networks[8], including exploration of communities identified via linkage-based clustering [15]. Others visualize personal e-mail archives with applications ranging from workplace productivity to personal reflection [14, 21, 22, 23].

Two systems highly relevant to our work on social topologies are Whittaker et al.'s ContactMap [25], and Fisher's Soylent [5].

ContactMap provides an editable visualization of personal contacts, spatially organized and colored by group membership. Akin to our SocialFlows interface, ContactMap allows contacts to be placed into multiple groups and mines a user's email archive to seed the display. Our work similarly is predicated on the insight that communication patterns provide rich information for mining nuanced social structures, but our system is unique in its ability to represent both overlapping and nested groups.

Moreover, ContactMap does not automatically suggest group structures, instead requiring manual layout and assignment of each contact. The designers of ContactMap attempted to automatically seed groups, but encountered difficulties, noting that *"users were not satisfied with these automatic techniques, arguing that they were neither intuitive nor useful for social communication tasks. Rather than ties of greater or lesser strength, users wanted to group contacts based on their affiliation, work project, or social category."*

We, too, have found that mining social groups that are acceptable to users is a challenging task. However, our largely positive user study results present evidence that the combination of automatic seeding and direct manipulation editing can improve the creation and management of acceptable social topologies.

Soylent comprises a visual analytic system designed to aid exploration of personal, social data with an end goal of incorporating social context into collaborative work. Soylent, too, uses email as the primary data source for social contact information. It provides several linked visualization views for data navigation which allow filtering and drill-down into the underlying data. A main goal of Soylent is to facilitate the discovery of different types of social groups based on collaborative behavior properties. Several group types are described in the paper.

Unlike SocialFlows and ContactMap, Soylent functions as a general analysis tool rather than an end-user system. Moreover, it does not focus specifically on the task of eliciting significant social groups. While users may explore social group structures within Soylent, discovering and maintaining group *identity* is not supported.

SOCIAL TOPOLOGY MINING ALGORITHM

Instead of asking users to manually craft their social topologies, we infer them automatically from users' online communication patterns. Our social topology mining algorithm takes, as input, the most recent 2 years' worth of a user's *sent mail* folder, and outputs a social topology. In this section, we first introduce the core concepts used in our algorithm. We then present the algorithm itself. Finally, we comment on our data filtering and preparation techniques.

Social Molecules

The concept of a *social molecule* is central to social topology structure. A social molecule is a group of people that comprise a logical social *unit*. Informally, we envision a social molecule as a set of people that co-occur as communication recipients so frequently that splitting them makes little sense. Note that an individual may belong to several social molecules: consider a Math study group and a group of college roommates, for example. While people might belong to both groups, each group's constituents perform a clear social role.

Given a set of emails, a natural proxy for its social molecules is the collection of its unique recipient sets. However, this approach will result in an unruly topology consisting of too many groups. The goal of our algorithm is to reduce the noise in the topology by eliminating minor subsets of groups and by merging highly overlapping groups. We refer to the former case as *identity subsumption* and the latter as *identity unification*.

Identity Subsumption

Let us first present the intuition behind identity subsumption. Given two groups of message recipients g_1 and g_2, $g_1 \subset g_2$, we need not include g_1 in the social topology if it receives relatively few messages compared to g_2. That is, g_1 loses its identity if it can be represented well enough by g_2. Subsumption of g_1 by g_2 introduces an inaccuracy in representation, which can be measured by the amount of information leaked were messages sent to g_1 shared with g_2. If this leak is below some threshold, we can subsume g_1 with g_2.

Let $\text{msgs}(g)$ be the number of messages received by group g. Any message received by a group is necessarily received by a subgroup, thus $\text{msgs}(g_1) \geq \text{msgs}(g_2)$, for all $g_1 \subset g_2$. For example, if 30 messages were sent to $g_1 = \{A, B, C\}$ and 20 messages were sent to $g_2 = \{A, B, C, D\}$, $\text{msgs}(g_1) = 50$ and $\text{msgs}(g_2) = 20$.

To measure information leak, we define the *sharing error* (serr) of group g_1 with respect to a superset g_2 as

$$\text{serr}(g_1, g_2) = \frac{(|g_2| - |g_1|) \times (\text{msgs}(g_1) - \text{msgs}(g_2))}{|g_2| \times \text{msgs}(g_1)}$$

The sharing error captures the ratio of non-recipients to the total message volume, should the subset be subsumed by the superset. The total number of non-recipients is the product of the number of messages sent only to the smaller group and the number of non-recipients in the larger set. The message volume is the product of the number of messages sent multiplied by the size of the larger group.

Suppose group $\{A, B, C\}$ received 103 messages, and 100 of these also include D.

$$\begin{aligned} \text{serr}(\{A, B, C\}, \{A, B, C, D\}) &= \frac{(4-3) \times (103 - 100)}{4 \times 103} \\ &= 3/412. \end{aligned}$$

Sharing error is low when g_1 is relatively similar to g_2 and a small fraction of messages are directed only to g_1. The identity of group g_1 is *subsumed* by g_2 if $\text{serr}(g_1, g_2)$ is less than the *self-identity threshold*, an algorithm parameter discussed below.

Identity Unification

Members of logically cohesive groups might not all appear together on any messages. For example, consider a research group of PhD students that spans several years. Graduated group members might never appear on an e-mail with new group members. However, we may wish to invite all the members to an alumnae party. In such cases, we create the overall group by synthesizing it from smaller subsets.

To measure the similarity between two groups, we use the Jaccard metric, a standard measure of set similarity. The Jaccard similarity of groups g_1 and g_2 is defined as:

$$\frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}$$

Two groups are unified if their Jaccard similarity exceeds a parameter, called the *similarity threshold*.

Algorithm Parameters and Default Values

Our algorithm is tuned via the following parameters:

1. *Minimum group size*: the number of people that must belong to a group for it to be part of the topology. As we are uninterested in individuals, the default value is 2.
2. *Minimum message count*: the number of messages in which a group must appear in the recipient list for the group to be considered significant. The default is 5 since only the last 2 years' worth of mail is considered.
3. *Self-identity threshold*: the maximum sharing error we are prepared to accept when one social molecule subsumes another. We have empirically found that a default threshold of 0.3 generates useful groups.
4. *Similarity threshold*: the minimum Jaccard similarity between two groups before being unified. The default similarity threshold is 0.35. This prevents the algorithm from combining two groups of two, with one person in common between them.

The Algorithm

Our algorithm comprises 3 phases. The first phase extracts relevant *social molecules*, according to the *minimum group size* and *self-identity threshold* parameters, from the input email corpus. Recall that a person may belong to multiple social molecules. This crucial property is what enables *overlapping* groups later on.

The second phase of the algorithm merges the social molecules from Phase 1 into larger groups. Merging is based on molecule co-similarity—controlled by the *similarity threshold*. The result of Phase 2 is a set of overlapping groups, each of which may or may not occur uniquely on a single message in the corpus.

The final phase of the algorithm creates a hierarchical structure of the topology for visualization.

Phase 1: Social Molecules and Identity Subsumption

Phase 1 extracts a relevant set of social molecules in the user's social graph. It outputs social topology, T, consisting of all the extracted social molecules.

1. *Initialize the topology.* Add each unique message recipient group, g, to T if $|g|$ is larger than the *minimum group size* threshold.
2. *Find all candidate social molecules.* Add to T pairwise intersections between all pairs of groups in T meeting the *minimum group size* threshold, until no more new intersections can be found. This generates the maximal subsets.
3. *Remove insignificant social molecules.* Remove from T any candidate set whose message count falls below the *minimum message-count threshold*.
4. *Subsume irrelevant social molecules.* To calculate identity subsumption, we arrange the social molecules in T in decreasing order of group size. We iterate through the list of social molecules, dropping all molecules that can be subsumed by any of the larger molecules that remain.

Phase 2: Manufacturing Supergroups

Phase 2 adds to the social topology cohesive supergroups derived from the social molecules generated in Phase 1 using identity unification. It iteratively creates new groups by combining the pair of groups that have the highest similarity across all groups inferred thus far, until this similarity falls below the *similarity threshold*. Note that some of the groups created by Phase 2 may be "manufactured" – that is, they do not exist in their entirety in any message in the original email corpus.

Phase 3: Organizing the Groups in a Hierarchy

Phase 3 of our algorithm constructs a hierarchy of social groups. It computes an ordering that prioritizes groups by their total number of received emails, while also placing similar groups in close proximity to one another.

We define the parent of a group g in the social topology T to be the proper superset that has the smallest sharing error with respect to g. The algorithm applies two sorting criteria to order groups at the same hierarchy level:

1. Significant groups should be displayed first. A group's significance is defined by its *group mass*, which is the sum of the messages received by each member in the group.
2. Groups similar to each other should be shown together as a unit to facilitate subsequent group merging and splitting operations by users. Two groups are considered similar if their Jaccard similarity is above a pre-defined threshold.

For each level of the hierarchy, the algorithm displays the most significant group and its children, followed by significantly similar groups. This process is repeated until all groups are assigned to the hierarchy.

Data Preparation

In this section, we detail our methods of filtering and cleaning our data before running it through the algorithm.

Sent mail. Although our approach would work on any email corpus, we restrict input to our algorithm to comprise only users' sent mail folders. Firstly, sent messages tend to be the best reflection of the user's ties. Sending email involves time and active effort on the user's part, so each outgoing message has a non-zero cost. Secondly, considering only sent messages avoids the problem of pollution due to spam. Finally, the number of messages sent to a person or group is representative of the relevance of that relationship to the sender. While we do not claim that e-mail frequency gives an ordering over relationship strength, we find empirically that the sent message count corresponds with tie strength at a coarse granularity.

Discounting messages over time. In initial studies on our own data, we found that social groups inferred from older messages were often less relevant than groups inferred from newer messages. In our current implementation, we truncate the email corpus to a timespan of 2 years. In future work we will investigate weighting functions over historical periods.

Entity resolution. Because email addresses change over time, identifying unique individuals in the input corpus is essential when producing a structurally correct social topology. Entity resolution of email addresses is an open research problem. Some current work attempts to use the social graph structure itself to perform sophisticated disambiguation of graph entities [2, 9]. We take a middle-ground approach by unifying all entries whose name or email address are equivalent. This is made possible by the fact that most email systems follow the RFC-822 syntax. We use case insensitive string comparisons for determining name equivalence, and are tolerant of common variations in naming patterns, such as "Firstname Lastname" versus "Lastname, Firstname". We find that this approach works well in practice on datasets encountered in our experiments.

Mailing list removal. Mailing lists are problematic for our algorithm on two levels: they tend to have high frequencies and it is difficult to determine explicit mailing list membership. Currently we simply remove all unambiguously identifiable mailing lists (addresses containing @googlegroups, @yahoogroups or @lists) from recipient lists in the input corpus. In the future, we would like to expand mailing lists into their membership sets.

SOCIALFLOWS: A SOCIAL TOPOLOGY BROWSER

In this section, we describe SocialFlows[2], a browser interface that allows users to explore and modify social topologies, as well as port final topologies for use in Gmail and Facebook.

Presenting Social Topologies

We visualize social topologies by showing mined groups and their constituent contacts. Contacts are represented by a compact grid of profile photos. Profile photos facilitate easy identification of a contact's membership across groups and provide a more engaging experience; prior visualization

[2]A video demonstration of the user interface is available at http://mobisocial.stanford.edu/socialflows/SocialFlowsDemo.mov

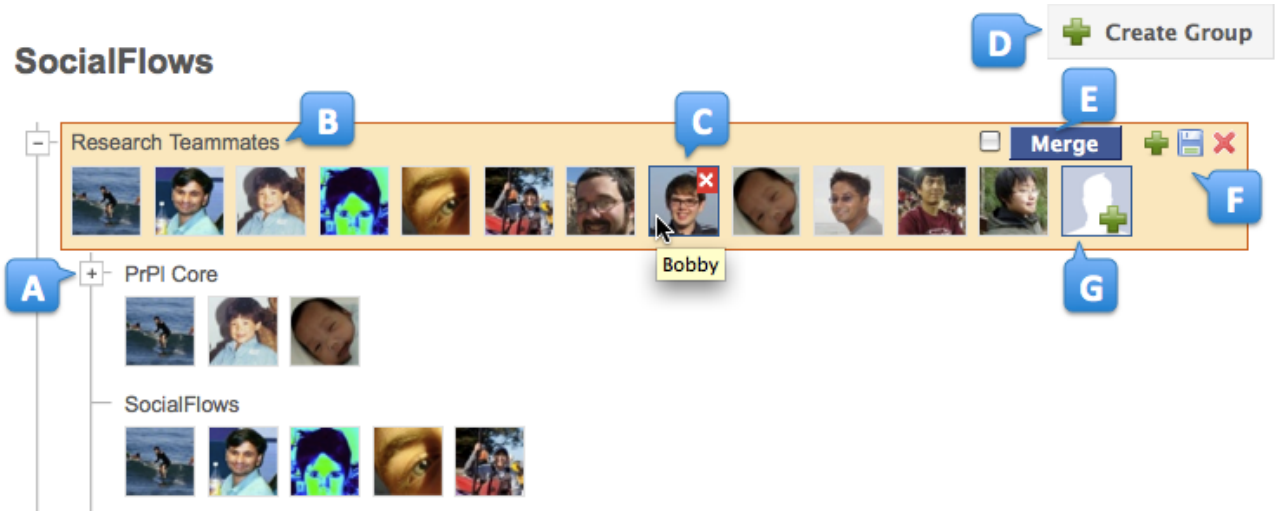

Figure 2. The SocialFlows user interface. Users can explore, browse and edit their social topology. Annotated points of interest highlight: (a) hierarchical nesting of subsets; (b) editable group labels; (c) tooltip and delete option on mouse hover; (d) new group creation; (e) group merge tools; (f) additional group editing tools; and (g) option to add a new contact.

research suggests that images are the primary visual cue by which users identify contacts and communities [8, 25]. A placeholder picture labeled with the contact's name is displayed for social contacts without a profile photo. The order in which social contacts are displayed is consistent across groups, sorted by descending frequency of correspondence. Contact names and details are revealed by a tooltip upon mouse hover and selecting a person also highlights all occurrences of that person in other groups. Hierarchical structure is depicted using an indented tree layout, akin to a file system browser. SocialFlows provides an expand-contract control (**A** in Figure 2) for each social group containing subsets.

Editing Social Topologies

The social topology automatically mined from email may not resonate precisely with the user's own mental model of social groups. Nonetheless, mined topologies can provide a very helpful starting point from which to refine one's model of personal contacts. To this end, SocialFlows allows users to edit their social topology via group composition and modification. Users can edit the composition of a social group by dragging and dropping contacts between groups, deleting a contact from a group by hovering over and clicking on the revealed X-mark (**C** in Figure 2), or adding a new contact to a group through the *Add a Contact* icon (**G** in Figure 2).

Users may also find it useful to consolidate social groups by merging two or more selected groups. SocialFlows facilitates this task by placing strongly similar groups near one another, as described under Phase 3 of the algorithm in the previous section. A user can then specify groups to merge by clicking on each respective group's Merge checkbox (**E** in Figure 2), and specify the destination group for the merged result by clicking that group's Merge button. All merged groups, except the destination group, are removed.

In addition, new top-level social groups can be created using the *Create Group* button (**D** in Figure 2). The per-group manipulation options (**F** in Figure 2) provide users with the ability to create a new child subset () out of an existing social group, save () a group, or delete () a current social group. Users can label their social groups via in-place text editing (**B** in Figure 2). Users can choose to port their saved groups to Gmail, as contact lists, or to Facebook as friend lists and groups.

SocialFlows also enforces invariants on the social landscape. For example, a parent in the social topology hierarchy is always a superset of its children and descendants and, conversely, children social groups are always subsets of their ancestors. This invariant is maintained in the user interface; users, for example, can add friends or social contacts to a child social group, and the added contacts are then percolated up the SocialFlows tree to each superset. Likewise, deleting a social contact from a parent social group results in that same contact being deleted from all subsets.

SOCIALFLOWS IMPLEMENTATION

As email is highly private, we package our email mining algorithm and SocialFlows as a downloadable application that runs on a user's own computer using Java Webstart. This lets our mining algorithm run over the email account that the user provides, as well as on locally stored mailbox archives. The application asks a user to provide an email account from which it can extract sent email messages (e.g., using POP3 or IMAP) and runs the social topology miner to deduce a user's social groups. The generated social topology is then presented in the SocialFlows browser.

Visualizing and exploring social topologies becomes a more engaging experience when friends and social contacts are represented by their photos instead of just their names. How-

ever, as most people do not have profile photos in their email address books, we have also made SocialFlows available as a web application that lets users supply their Facebook credentials through Facebook OpenGraph [3]. This enables SocialFlows to integrate social contacts from a user's email with the user's Facebook friends and their respective profile photos. Matching is done using a name-matching heuristic, and if a match is not found, a placeholder profile picture labeled with the contact's name is displayed instead.

Note that only the resulting social topology is provided to the SocialFlows web application for presentation to the user. All intermediate results are confined to the user's local machine, thus preserving the privacy of the user's email.

USER EVALUATION OF SOCIAL TOPOLOGIES

To assess the effectiveness of our approach, we conducted a user study comparing our SocialFlows system to contact groups in Gmail which is a reasonable representation of state of the art approaches to online social contacts organization. The goal was to assess the degree to which our automatically computed social topologies assisted users in managing their social contacts. We hypothesized that not only would users find it easier to create social topologies using a seeded topology generated by our algorithm, but also that they would find these topologies more useful than those constructed from scratch. We were also interested in general assessments of our interface.

Methods

Social groupings are highly contextual, and so evaluations of generated topologies may depend upon the task at hand. Simply presenting a topology and asking users whether or not it is "good" may not be suitably informative. In our study, we attempted to control situational context by giving users a set of prompts in the form of concrete information-sharing scenarios (described subsequently). Based on these prompts, users were asked to construct contact lists that they might use within the given task scenarios. Each user was asked to perform this task twice: once from "scratch," with no suggested topology structure, and once using an editable topology generated from their e-mail data using our algorithm.

Our study used two interface conditions: Gmail's Contact Manager [7] and SocialFlows. In the first condition, we asked users to design a partial social topology from scratch using Gmail's contact group management interface. In the second task, we asked users to build a social topology using our SocialFlows interface with a topology seeded by our algorithm. Both interfaces allow users to create and delete *overlapping* social groups, and edit social groups by adding and removing recipients from a contact list. The Gmail interface represents contact groups as a flat list of groups, whereas the SocialFlows interface represents proper subsets hierarchically. We randomized interface presentation order across subjects. Subjects were given a couple of minutes to navigate each interface before starting a task; once started, subjects were given a maximum of 15 minutes to complete each task.

Due to privacy constraints, we did not observe participants' final topologies directly. Instead, we collected participants' feedback as well as descriptive statistics about the topologies in order to perform our evaluation. We measured how long it took subjects to complete a task, and upon completion asked them to state why they stopped (e.g., task completed successfully, ran out of time etc.). At the end of the study, we asked subjects to rank the *efficiency* of each interface and the *usefulness* of the topologies, assuming they were able to export them to other online services. Finally, we encouraged them to comment openly on their experiences.

We chose to use the Gmail interface as a control (rather than Facebook, for example) for several reasons. First, Gmail and SocialFlows have identical contact sets, which is necessary for fair comparisons between tasks. Second, Gmail's contact group management interface is representative of the state of the art in online contact organization, which gives our results ecological validity than a simple comparison of "from scratch" vs. "from template" conditions in the SocialFlows interface only.

19 subjects participated in the study. Most participants were students in our computer science department. In pilot studies, we found that results varied widely depending on two factors: (1) whether the e-mail data input to our system came from the user's *primary* e-mail account, and (2) the volume of e-mail data input to the algorithm. All datasets were truncated to contain messages that had been sent in the last 2 years only; however, we found that users with relatively small sent mail folders ($<$ 1,000 messages) got sporadic results. Combined with our decision to use the Gmail interface as a control, we limited the study to users who used Gmail as a primary e-mail account, and had more than 1,000 messages in their sent mail folder.

In all cases, our email analysis algorithm ran on users' own machines, typically their laptop computers. The algorithm itself is fast, taking less than a second to complete for all users. Most of the waiting time was to connect to the email server and fetch message headers.

Task Prompts

To contextualize the study tasks, we developed four sample information-sharing questions. We asked subjects to build a partial social topology based upon the answers to these (and similar) questions. *Before* attempting any tasks, we presented subjects with a clear definition and examples of a social topology. We then informed them that their topology should make answers to such questions composable, as opposed to creating four groups that explicitly answer the given questions. We used the following prompts:

1. After a long time away, you are returning to town where you have several good friends (e.g., the town where you grew up). Unfortunately, you will be in town for a few days only, and you will have time to see just a few, essential people. Who would you inform of your travel plans?
2. You're turning 21 and want to throw a party. Who would you invite?

	Avg.	Min	Max	Std. Dev.
Initial Groups	476.7	104	920	251.5
Subsumed Groups	82.8% (398.9)	2.7% (13)	98.4% (872)	23.6% (244.4)
Unified Groups	3.3% (17.1)	0	32.3% (155)	7.2% (35.3)
Final Groups	20.5% (94.8)	1.6% (2)	129.4% (624)	30.0% (143.6)
Root-level Groups	5.2% (27.4)	1.6% (2)	15.0% (129)	3.2% (28.0)

Table 1. Summary statistics of social topologies generated by our algorithm from subjects' email data.

3. You have a few extra tickets to an event (e.g. basketball game, concert) that you're really excited about. Assuming that you cannot sell the tickets, who would you invite?

4. You have just been diagnosed with a medical condition (sufficiently severe that you would like to maintain some confidentiality, but not top secret). Who would you share this information with?

We designed these questions to be general enough to apply to a broad range of users, but specific enough to target delimited social groups and varying tie strengths. We expected the questions to elicit overlapping social groups. Finally, we were careful to design questions that would present realistic scenarios to most users. However, we do note that the questions assume that most users have some assortment of family, friends and coworkers in their social networks, and communicate with them via email.

Results

Of the 19 subjects, 10 used the Gmail interface first and 9 used SocialFlows first. As we found no statistically significant differences between the response distributions conditioned on presentation order, we focus our discussion on the pooled results only. We summarize the behavior of our algorithm (used in the SocialFlows interface) in Table 1. Table 1 shows that on average individuals write to close to 500 different groups in a course of 2 years. Presenting all such groups in a list to the user would not be useful. Our algorithm reduces the *total* number of groups by a factor of 5; when organized into a hierarchy the reduction is even smaller, and the root groups typically comprise 5% of the number of original recipient sets. On average, then, final topologies contain approximately 100 total groups, with approximately 30 root groups.

Interface	Out of Time	Satisfied	Too Hard	Bored
Gmail	21% (4)	42% (8)	26% (5)	11% (2)
SocialFlows	21% (4)	74% (14)	5% (1)	(0)

Table 2. Task stopping reasons.

Table 2 and Figure 3 summarize subjects' reasons for completion and completion times, respectively. Although users were told that they should take only 15 minutes to complete each task, several users were disinclined to stop after the time limit. Of the 19 users, 6 users abandoned the task based on the Gmail interface midway, citing that it was too unwieldy to construct their envisioned social topologies. To assess the significance of the time to completion results, we used a paired Wilcoxon sign rank test, modeling task abandonment as "infinite"[3]. We find that performance using SocialFlows is significantly faster ($p = 0.006$).

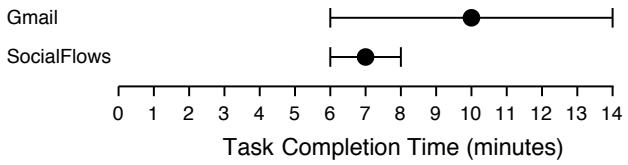

Figure 3. Median task-completion times (minutes). Error bars show median absolute deviations (MAD). The Gmail result is based on only 13 of the 19 users, as 6 found the task intolerable and abandoned it.

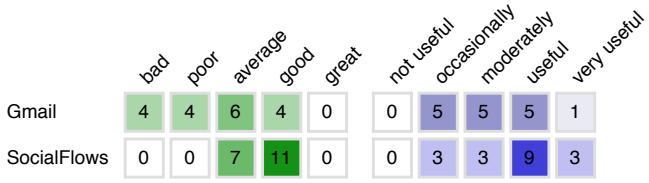

Figure 4. User study ratings. (a) Ease of social topology construction. (b) Perceived usefulness of topology for online sharing tasks. Note that not all counts match, as some subjects chose not to respond.

We note that most users (74%) stopped creating their SocialFlows topologies because they were satisfied with the result; comparatively fewer users (42%) were satisfied with their Gmail topologies upon task completion. A chi-square test of these counts indicates significance at the $p < 0.1$ level, but not at $p < 0.05$. This is unsurprising, given the low number of cell counts. Binning responses into "satisfactory" and "unsatisfactory" (out of time, too hard, and bored), we find that SocialFlows results in significantly more satisfactory outcomes ($\chi^2(1,38) = 3.886$, $p = 0.049$). One explanation noted by subjects is that drawing one's social topology from memory is difficult: people often forget to add individuals, or construct links between certain groups. Another explanation is that having a ready-made framework significantly reduces the cost of social topology construction.

Figure 4(a) presents users' ratings of the efficiency of social topology creation in each interface. Users were asked to provide these rankings after completing both studies. As we did not use a true Likert scale for these rankings, we treat the results as ordinal and use a paired Wilcoxon sign rank test for significance (we note that running an ANOVA instead,

[3]We used a non-parametric test as the times are not normally distributed. The test operates on ranks, so "infinite" times are handled appropriately.

treating responses as values from 1-5, gives similar results). We found that SocialFlows was rated as significantly easier to use ($p = 0.007$). As the Gmail interface represents the current state of the art in contact organization, this result argues that the SocialFlows interface provides a superior mechanism for creating social topologies. We acknowledge that our results do not indicate how much of the efficiency gain is apportioned to the *interface* versus the *algorithmically-seeded* groups; however, both user comments and user rankings of group quality, presented in Figure 4(b), support the hypothesis that a large portion of the efficiency gain is, in fact, due to the seeded topology.

To rate the quality of social topologies, we asked users to rank the perceived usefulness of their topologies under the hypothetical context that groups from their topologies could be exported to online services such as Facebook. The results are presented in Figure 4(b). We again use a paired Wilcoxon sign rank test on the ordinal rankings, finding that the differences between the Gmail and SocialFlows interfaces do not rise to a level of statistical significance ($p = 0.159$). However, we do not believe that this is sufficient to *reject* the alternate hypothesis that SocialFlows topologies are more useful. First, the SocialFlows responses do have a larger proportion of high scores. Second, 55% of participants stated that a major motivation for using the SocialFlows interface would be the *quality* and *correctness* of the groups.

User Impressions of the Gmail Task

Several users registered dislike of the Gmail task, citing tedium and lack of intelligent group suggestion as major drawbacks. Another strongly cited aspect of the Gmail task was the difficulty of creating a social topology from scratch. One user noted that imagining the groups themselves was easy, but deciding or remembering who should belong to which group was extremely difficult: *"I didn't like how I had to add each individual to each group individually; for instance, I had to imagine a group from scratch, and then I'd try and remember who should go in it, and then I'd forget people and want to go back, etc."* Similarly, another user noted that he/she had many friends, but *"within that broad group there is a range of comfort in communicating and trust,"* which suggests that granularity within groups is a natural phenomenon. In summary, users were generally unhappy about having to create topologies from scratch, complaining that not only is the task tedious, but also that it is difficult to remember who should be in which group.

User Impressions of the SocialFlows Task

Many users found the SocialFlows task interesting and novel, and users' comments about the task were generally positive, tending to focus on group quality. We note that some degree of novelty bias was likely present. 55% of users rated the correctness and accuracy of the groups as a highly redeeming feature of the algorithm; another highly ranked feature (also at 55%) was the *granularity* of the social groups. Despite the praise for the algorithm's correctness, many also noted incomplete or missing groups. This was not surprising to them; they explained that the missing groups consisted of individuals that they did not usually contact via email, or whom they contacted using a different email account. Several users suggested sourcing data from multiple email accounts, social networks, and even from telephone records to achieve improved results (c.f., [25]).

Users also provided helpful comments for improving the interface. Some noted that using small photographs as icons for contacts is sometimes confusing. A few users felt that there were too many subgroups within root groups, indicating that the algorithm's results may be too granular. Perhaps the most interesting take away from these comments is that users are able to identify *incorrect* aspects in a social topology quickly and accurately, despite their inability to easily create social topologies from scratch. Regardless of the efficiency gains, this alone is a motivation for initializing social topologies from existing data, as users can create reasonably accurate topologies by making small corrections to, or paring down, algorithmically generated topologies.

Summary

Our user study results are promising. User experiences with the SocialFlows interface were both significantly *faster* and rated significantly *more efficient* than their experiences with the Gmail interface. While users' rankings of perceived topology usefulness were not significantly different, qualitative responses argue that users found SocialFlows topologies to be of higher quality. We hypothesize that the algorithmically generated scaffolding provided in the SocialFlows interface decreases the overhead cost of social topology construction by allowing users to edit an existing topology instead of having to build it up from scratch. Additionally, we suspect that the scaffolding serves as a helpful prompt for articulating social group structures that are *naturally* nested and overlapping. As the Gmail contacts editing interface is reasonably representative of the state of the art, our results provide evidence that our contribution of the social topology structure, a generative algorithm, and our interface provide a much improved means of organizing a social landscape.

CONCLUSION

In this paper, we introduced *social topologies*: sets of possibly overlapping and nested groups that model the structure and content of a person's social ties. Noting that email accounts contain valuable data about one's social affiliations over time (c.f., [25]), we described an algorithm that derives social topologies from sent email messages. We then presented our SocialFlows interface with which users can explore, modify and port their generated social topologies.

In our user study, subjects found our SocialFlows interface to be significantly more efficient than a state-of-the-art interface for social contacts organization, Gmail's contact manager. In fact, over half of the subjects who used the SocialFlows interface first (5/9) chose to quit rather than complete the task using the Gmail interface due to the increased effort. Users' qualitative feedback similarly expressed appreciation for the ability to quickly modify generated groups in our interface. The study results support our hypothesis that automatic generation of social topologies, which users

may then modify, improve users' ability to create and maintain models of their social groups.

While in this paper we focused on email mining for the purposes of developing and supporting a contact management user interface, we believe that social topologies have broader analytical applications. In particular, we know of no work in social network analysis that focuses explicitly on creating hierarchical groupings that overlap to the extent that ours do, despite the fact that this property follows naturally from real-life social structure. Continuing research along these lines is crucial: as socially-oriented online services become increasingly prevalent, we anticipate that improved means of organizing and maintaining our personal, social landscapes will become absolutely necessary.

REFERENCES

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. SIGMOD 1993*, pages 207–216.
2. A. Culotta, R. Bekkerman, and A. Mccallum. Extracting social networks and contact information from email and the web. In *Proc. CEAS 2004*.
3. Facebook. Facebook for websites. Retireved from: http://developers.facebook.com/docs/guides/web.
4. T. Falkowski, J. Bartelheimer, and M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *Proc. WI-IAT 2006*, pages 52–58.
5. D. Fisher and P. Dourish. Social and temporal structures in everyday collaboration. In *Proc. CHI 2004*, pages 551–558.
6. E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proc. CHI 2009*, pages 211–220.
7. Gmail. About the Contact Manager – Gmail Help. Retrieved from: http://mail.google.com/support/bin/answer.py?hl=en&answer=77259.
8. J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proc. InfoVis 2005*, pages 33–40.
9. R. Hölzer, B. Malin, and L. Sweeney. Email alias detection using social network analysis. In *Proc. 3rd International Workshop on Link discovery 2005*, pages 52–57.
10. G. Kossinets and D. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88, 2006.
11. J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of large social and information networks. In *Proc. WWW 2008*, pages 695–704.
12. A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, 2007.
13. J. Moody and D. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68(1):103–127, 2003.
14. C. Neustaedter, A. Brush, M. Smith, and D. Fisher. The social network and relationship finder: Social sorting for email triage. In *Proc. of CEAS 2005*.
15. M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133:1–5, 2004.
16. G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.
17. Radicati. The Radicati Group, Inc. Releases Q2 2008 Market Numbers Update, August 2008. Retrieved from: http://www.radicati.com/?p=638.
18. M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *Proc. KDD 2010*, pages 233–242.
19. M. Siegel. Zuckerberg: "Guess What? Nobody Wants To Make Lists", August 2010. Retrieved from: http://techcrunch.com/2010/08/26/facebook-friend-lists/.
20. J. Tyler, D. Wilkinson, and B. Huberman. E-mail as spectroscopy: Automated discovery of community structure within organizations. *The Information Society*, 21(2):143–153, 2005.
21. G. D. Venolia and C. Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proc. CHI 2003*, pages 361–368.
22. F. B. Viégas, D. Boyd, D. H. Nguyen, J. Potter, and J. Donath. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *Proceedings of HICSS*, 2004.
23. F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Proc. CHI 2006*, pages 979–988.
24. S. Wasserman and K. Faust. *Social network analysis: Methods and applications*. Cambridge University Press, 1994.
25. S. Whittaker, Q. Jones, B. A. Nardi, M. Creech, L. Terveen, E. Isaacs, and J. Hainsworth. ContactMap: Organizing communication in a social desktop. *ACM TOCHI*, 11(4):445–471, 2004.
26. D. Zhou, E. Manavoglu, J. Li, C. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proc. WWW 2006*, pages 173–182.