An Algorithm and Analysis of Social Topologies from Email and Photo Tags

T. J. Purtell   Diana MacLean   Seng Keat Teh
Sudheendra Hangal   Monica S. Lam   Jeffrey Heer

Computer Science Department
Stanford University
Stanford, CA 94305
{tpurtell, malcdi, skteh, hangal, lam, jheer}@cs.stanford.edu

ABSTRACT

As peoples' participation in social media increases, online social identities accumulate contacts and data. We need a mechanism for creating a succinct but contextually rich representation of a person's "social landscape" that would facilitate activities such as browsing personal social media feeds, or sharing data with nuanced social groups.

We formulate the social topology extraction problem as the compression of a group-tagged data set in which each group has a significance value, into a set containing a smaller number of overlapping and nested groups that best represent the value of the initial data set. We present four variants of a greedy algorithm that constructs a user's social topology based on egocentric, group communication data. We analyze our algorithm variants on about 2,000 personal email accounts and 1,100 tagged Facebook photograph collections. We find that our algorithm variants produce different topologies suitable for different purposes.

We show that our algorithm can capture 80% of the input data set value with 20% and 42% of the number of input groups for email and photographs respectively. Using edit distance as an objective metric, we also show that our algorithm outperforms results generated by Newman's modularity-based clustering algorithm. We conclude that our algorithm is appropriately designed to find significant groups of friends from social contact data.

1. INTRODUCTION

While millions of users have accumulated large lists of "friends" in online social networks, managing these flat lists is challenging. A natural organizing principle is to assign friends to different categories that can then be used for targeted sharing and filtering of social content. However, existing tools such as Facebook friends lists or Gmail contact groups require users to create these lists from scratch, making this a tedious task that is not done by most users [20]. To address this problem, we present a novel algorithm for compressing a user's communication data into a compact set of relevant groups that may be useful for the aforementioned tasks.

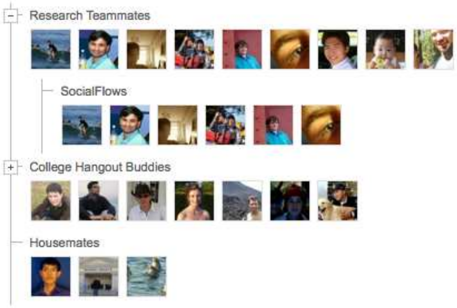


Figure 1: An example of a social topology.

1.1 Social Topology

MacLean et al. have proposed the concept of a *social topology* – the structure and content of a person's social affiliations, comprising a set of overlapping and nested groups – as a first-class structure for facilitating social-based tasks such as data sharing or digital archive browsing [13]. We exploit the observation that a user's social topology is captured implicitly in routine communications, photographs, and others forms of personal data. In this paper, we present a novel algorithm for generating a social topology from a user's grouping data, assuming a constraint on the size of a social topology. We define group communications data as corpora in which items may be tagged with more than one social identity; for example, email, tagged photograph collections, and co-location data.

Figure 1 shows several defining properties of a social topology. First, any individual in the topology may appear in several groups. This models people who play several roles in the user's life, such as being both a colleague and a friend. Second, it may contain *manufactured* groups – group of people who never occur together in a single item in the original data set. Consider a university lab whose membership changes annually: a "core" group, such as a faculty team, might persist in the group over time, but never appear uniquely in a photograph collection. Third, social topology groups may

The 5th SNA-KDD Workshop '11 (SNA-KDD'11), August 21, 2011, San Diego, CA USA.
Copyright 2011 ACM 978-1-4503-0225-8 ...$5.00.

be *nested*. This captures specific subgroups within a supergroup, such as siblings within a family. Finally, a "group" may consist of just a single individual who is sufficiently important.

We formulate the problem of deriving a social topology as follows: given a data set d of group communications data, a value function v which measures the significance of a group with respect to d, and a budget of b groups, find b groups whose aggregate value is maximized. We derive these groups only from the data that is directly visible to the user, making these groups ego-centric.

1.2 Contributions

The contributions of this paper include:

- A greedy algorithm for constructing social topologies from group communications data. The algorithms make different trade-offs and can be tuned based on the target application. Our algorithm is incorporated in a Facebook application called GroupGenie[1].
- A validation and comparison of our algorithm using two data sets: a collection of 1,995 personal email archives containing over 24 million sent email messages and a set of 286,038 tagged photos from 1,099 Facebook users.
- An evaluation and comparison of social topologies constructed from these data sets. The evaluation includes a comparison with Newman's clustering algorithm using edit distances as an information-theoretic metric.

Source code for the algorithm is also publically available[2].

2. RELATED WORK

There is a substantial body of work in analysis of social data, both for global (e.g., [9, 11]) and ego-centric (e.g., [4, 7, 14]) networks. Below, we discuss and contrast prior work with our approach.

Clustering algorithms aim to elicit communities from a graph structure. Traditional algorithms based on hierarchical agglomerative clustering *partition* the input graph, disallowing node overlap between clusters [3, 17]. We find this approach unsuitable for our purposes, as one person can adopt several social roles simultaneously.

Palla et al. present an algorithm that discovers overlapping communities in global, unweighted networks [18]. Communities are generated in a bottom-up fashion from k-cliques. Taking a different approach, Banerjee et al. introduce "model based clustering", a probabilistic graphical model for inferring overlapping clustering [2]. Huberman et al. extract overlapping social clusters by running an edge betweenness clustering algorithm several times, starting from a network where an unweighted edge exists between 2 people if 5 or more messages were exchanged between them [23]. Lancichinetti et al. present another method for detecting overlapping and hierarchical structure in complex networks [10].

There are three major differences between our work and the above algorithms. First, these algorithms make the assumption that the global structure of the network is available. Second, many of them are evaluated on networks formed by publicly available information, while we evaluate our algorithm on personal data, where there may be different patterns of group formation. Third, the input model of the graph is reduced to edges between individuals, ignoring the fact that the input data was grouped in the first place.

Visualization and interface techniques such as ContactMap [24], Vizster [8] and LinkedIn InMaps [12] help users view and organize their social networks. Previously published work by MacLean et al. describes an algorithm to derive overlapping and hierarchical groups, and an interface to edit those groups [13]. This algorithm required the use of several parameter settings and was evaluated in a smaller study involving email data sets of 19 users; moreover it does not seamlessly handle individuals. In contrast, the work reported in this paper presents an algorithm with better accuracy, and has been evaluated on a larger scale on multiple data sets.

Association rule mining is a technique for finding related item sets in a corpus, given a specific seed [1]. Roth et al. present a group-finding algorithm for Gmail in which the goal is to complete the group as accurately as possible given an initial seed [19]. Like us, they assume that communications reflect *implicit* social structure, and use communication frequency as a proxy for tie strength. They develop an *interactions rank* metric that gives an ordering over unique recipient groups, allocating points according to communication frequency, recency, and direction. However, seed-based approaches are generally inadequate for the purposes of helping users construct a social topology; for example, Gmail users cannot access the set of probable groups or use them for other purposes. As a result, the algorithm does not directly create a summary of the input groups.

Graph summarization techniques are often applied to the problem of web graph compression. A small portion of graph summarization research focuses specifically on reducing the size and complexity of network data. Tian et al. present two approaches: SNAP, for lossless compression and k-SNAP, for lossy compression [22]. Taking an information-theoretic approach, Navlakha et al. employ the minimum description length (MDL) principle to produce a graph summary and a list of "corrections", allowing for both compression and perfect reconstruction of the input graph. The graph may be optionally reduced if lossy compression can be permitted [15]. The same authors employ this method to obtain rich but manageable summaries of protein interaction networks [16].

3. ALGORITHM

Our goal is to derive a user's social topology, consisting of potentially overlapping and nested groups of friends, from a corpus of a user's group communication. Our algorithm is parameterized to find the most significant *given* number of groups.

3.1 Problem Statement

We define a social topology to be a set of unique, potentially overlapping and nested groups, each of which has some *value*, and each of which is comprised of members drawn from the user's global set of friends. Permitting nested groups lends increased granularity to the topology, while permitting overlapping groups allows us to represent people who play multiple roles in the subject's life. Intuitively, the value of a group reflects the proportion of information that the user chooses to share with it, and we consider groups with a higher information share to be more important than others. We generate social topologies from a single user's *ego-centric*

[1] http://mobisocial.stanford.edu/groupgenie

[2] https://github.com/mobisocial/groupgenie-algo/

grouping data such as email records or tagged Facebook photographs.

Ego-centric group communication datasets already contain a natural social topology: the unique groups that occur together on items in the data set. Each of these groups can be assigned some appropriate valuation. For example for a user's collection of sent email, the natural social topology would be the unique recipient sets in the data set, and the value of each recipient set might be a function of its size and the number of messages on which it appears. Therefore our task of social topology construction is a task of *compression*, in which we want to reduce the natural social topology into a manageable size, while maximizing its value. We may need to combine groups in various ways, as well as drop groups from the topology altogether, if needed. Our problem formulation requires that each group in the original social topology be represented by at most one group in the compressed topology.

Depending on the objective, there are different trade-offs in generating a social topology. For example, we may wish to create a social topology that includes mostly *core* persons from different facets of our lives. Alternatively, we may wish to create a social topology containing as many related people as possible. In order to accommodate such diverse objectives, we introduce the notion of a *value function* that evaluates the value of each group in the generated social topology based on its mapping from the original one.

The social topology compression problem is thus defined as follows. Given

- a set of friends F,
- a natural social topology S consisting of unique groups $g \subset F$, where the value function $v_0(g)$ denotes the significance of the group g,
- a size b which is our *budget*, or number of groups required in the final topology,
- a value function $v(g, r)$, where g is the representative for a set of groups $r \subseteq S$.

find a social topology S' and a representation map R, mapping each $g \in S'$ to a non-overlapping set in S, such that $\sum_{g \in S'} v(g, R(g))$ is maximized.

3.2 The Sharing Value Metric

Our value function is based on a model of information sharing and over-sharing. Intuitively, if group g in the original social topology maps to group g' in the final social topology, the value is high if g' has the same members and low if g' has many additional members. The ratio between the number of common elements to the size of g determines the fraction of the positive contribution of g's value to g'. For each friend in g' and not in g, information from g is over-shared. Since different uses of social topologies may desire different over-sharing penalties, we allow the algorithm be parameterized with a penalty weighting function $w(f, g)$ that determines the penalty to be applied to each unit of over-sharing with friend f not in group g. We thus define a value function based on information sharing as

$$v(g', r) = \sum_{g \in r} \frac{v_0(g)}{|g|} \left(|g' \cap g| - \sum_{f \in (g' - g)} w(f, g) \right),$$

where $w(f, g)$ is the over-sharing penalty to be applied to friend f for group g.

One possible penalty weighting function is a simple constant, i.e.,

$$w(f, g) = C$$

If C is 0, there is no penalty for over-sharing; if C is 1, every person that a data item is overshared with costs as much as the value contributed by a person who was in the original group that the item was shared with.

A more sophisticated approach is to use a weighting function that depends on the relationship between the original group and the friends an item was over-shared with. Friends who are not in the original group g, but participate with members in group g in other groups, should have a lower sharing penalty. Let $P(\overline{f}|f')$ denote the conditional probability of not finding f in groups containing f'. Then we can define a function for the over-sharing penalty weight as

$$w(f, g) = \frac{1}{|g - \{f\}|} \sum_{f' \in (g - \{f\})} P(\overline{f}|f')$$

3.3 A Greedy Algorithm

We define a set of permissible actions, called *moves*, that may be taken on groups in a social topology. All moves reduce the social topology size by 1 and reduce the value of the topology according to its *error function*. Starting with the natural social topology, our algorithm greedily picks the move that maximizes the value of the resulting social topology until the topology is reduced to the desired size b.

We define the initial representation mapping R to simply map each group g to itself; if g is a group in the original topology, $v(g, \{g\}) = v_0(g)$. The moves and their error functions are defined below.

- DISCARD. Discard a group from the topology, thus losing the group's entire value.

$$E_{\text{DISCARD}}(g, r) = v(g, r)$$

- MERGE. Merge two groups to create a union that inherits the combined value, appropriately penalized to account for their membership mismatch. The over-sharing penalty built into the value metric ensures that the most closely related groups have the lowest error.

$$E_{\text{MERGE}}(g_1, r_1, g_2, r_2) = v(g_1, r_1) + v(g_2, r_2) - v(g_1 \cup g_2, r_1 \cup r_2)$$

- INTERSECT. Intersect two groups to capture the importance of a shared subset.

$$E_{\text{INTERSECT}}(g_1, r_1, g_2, r_2) = v(g_1, r_1) + v(g_2, r_2) - v(g_1 \cap g_2, r_1 \cup r_2)$$

- TRANSFER. Transfer the representation of a second group to the first group; the second group is discarded, but its value is partially transferred to the first, taking into account the over-sharing penalty.

$$E_{\text{TRANSFER}}(g_1, r_1, g_2, r_2) = v(g_1, r_1) + v(g_2, r_2) - v(g_1, r_1 \cup r_2)$$

3.4 An Approximate Algorithm

In the algorithm above, each group's value is defined in terms of the values of the original groups they represent. To simplify the algorithm, we adopt the model where each member in a derived group contributes equally to the group's value. We can thus approximate the value of a derived group with a single quantity and compute the error term for each move based on the approximate value of its operands. The approximate value function is defined as

$$\overline{v}(g) = \begin{cases} v_0(g), & \text{if } g \in S, \\ \overline{v}(g_1) + \overline{v}(g_2) - E_m(g_1, g_2), & \text{if } g = m(g_1, g_2) \end{cases}$$

where $m(g_1, g_2)$ is the result of applying move m to g_1 and g_2. (Unary moves like discard are similarly defined.) Since in this model, the value of a group is considered uniformly distributed across all its members, the over-sharing error simply depends on the ratio of additional people getting the information to the size of the group. The error functions are thus analogously defined as:

$$E_{\text{DISCARD}}(g) = \overline{v}(g)$$

$$E_{\text{MERGE}}(g_1, g_2) = \sum_{f \in g_2 - g_1} \tfrac{\overline{v}(g_1)}{|g_1|} w\left(f, g_1 \cup g_2\right) + \sum_{f \in g_1 - g_2} \tfrac{\overline{v}(g_2)}{|g_2|} w\left(f, g_1 \cup g_2\right)$$

$$E_{\text{INTERSECT}}(g_1, g_2) = \sum_{f \in g_1 - g_2} \frac{\overline{v}(g_1)}{|g_1|} + \sum_{f \in g_2 - g_1} \frac{\overline{v}(g_2)}{|g_2|}$$

$$E_{\text{TRANSFER}}(g_1, g_2) = \sum_{f \in g_1 - g_2} \tfrac{\overline{v}(g_2)}{|g_2|} w\left(f, g_1 \cup g_2\right) + \sum_{f \in g_2 - g_1} \tfrac{\overline{v}(g_2)}{|g_2|}$$

The experimental results presented in this paper are based on this approximate algorithm, as summarized in Algorithm 1.

Algorithm 1 compressTopology(S)

```
Input   Initial topology $S = \{g_i\}$,
         a set of unique groups and values $\overline{v}(g_i)$.
Input   A budget $b$, the size of the final topology.
Output  $S$, the final topology

while $|S| > b$ do
  $g^* \leftarrow m(g_1, g_2)$ where $m$ is the lowest loss move $\forall g \in S$
  $\overline{v}(g^*) \leftarrow \overline{v}(g_1) + \overline{v}(g_2) - E_m(g_1, g_2)$
  $S \leftarrow S + g^* - g_1 - g_2$
end while
```

4. EXPERIMENTAL EVALUATION

We have evaluated different versions of our algorithm on 2 types of datasets, one using personal email archives, and another using Facebook photo tags.

4.1 Email

Our email dataset is comprised of email headers from 1,995 users' personal email archives, totaling over 24 million sent email messages. The dataset, provided by Xobni Inc., was collected from a subset of users of their Xobni Cloud service. The data we received was fully anonymized; all personally-identifiable information had been removed. Most of these users connected to Xobni via Outlook, so we estimate that much of the email activity may be work related. Figure 2 outlines statistical properties of the corpus. We restrict our algorithm input to *sent* email only, noting that this is a more accurate signal for social importance, as sending an email incurs a cost on the user, whereas receiving one does not [13]. This also has the advantage of excluding spammers and advertisers. We see some startling anomalies in the data set, such as an individual who sent as many as 160,000 messages, and a message addressed to 2,826 recipients! Note that the majority of messages are sent to only one person.

	Messages	People	Groups	Group Size
Lower Q.	2038	329	373	1
Median	6640	738	1104	1
Upper Q.	14684	1422	2451	1
Max	159697	20813	24306	2825
Mean	11521	1109.5	1814.9	1.5
Std Dev	15205.1	1328.6	2231.4	2.3
Total	24228571	2213486	3816668	35781399

Figure 2: Summary of the 1,995-person email data set.

	Photos	People	Groups	Group Size
Lower Q.	31	28	19	1
Median	106	62	54	2
Upper Q.	325	130	142	3
Max	3062	594	1050	111
Mean	260.3	90.9	109.6	2.4
Std Dev	392.2	88.8	143.7	2.8
Total	286038	99910	120457	682126

Figure 3: Summary of the 1,099-person photo data set.

4.2 Tagged Photos

Just as emails capture co-occurrence of recipients on mails, tagged photographs capture physical co-occurrence of subjects. Given the fact that photo sharing is one of the most popular forms of online social activity, tagged photographs are an excellent source of social topology data. To evaluate our algorithm on tagged photos, we have developed GroupGenie, a Facebook application that allows Facebook users to infer their social topology from their tagged photo data.

GroupGenie users have found the social groupings suggested to them by our algorithm helpful for both data sharing and communication tasks, and for a certain degree of personal self-reflection. An informal pilot study of about 30 users aged 17-19 found that the groups suggested to them were good enough, with a few minor edits, to publish to their profile pages as Facebook Featured Friends [6]. Some found it useful to use their groups in Facebook Chat [5] to do group-wide chats.

At the time of this writing, 1,099 Facebook users have used GroupGenie. Most of these users discovered GroupGenie through friends, and from a press article about an earlier version of this work [21], suggesting strong interest among Facebook users in tools to help them create groups.

Figure 3 provides summary statistics of the tagged photograph corpus. Note that the owner of the Facebook account, if present, is excluded from the input groups. There are distinct differences between this data set and the email data set in terms of group density. In particular, the average group size in a photograph is 2.4, compared to the average number of recipients on an email, which is 1.5. Moreover, more than half of the photographs are tagged with at least 2 people excluding the user; in contrast, a majority of emails involve only one other person excluding the user. On the other hand, tagged photograph collections are typically smaller than email collections, presumably due to the larger effort required to take, upload and annotate photographs.

5. ANALYSIS OF EMAIL DATASET

We run our experiment on four variants of our algorithm:

DISCARD. Considers only discard moves. This straw-man version simply reports the top b initial valued groups for a given budget b.

MERGE. Considers discards and merges, with a simple fixed penalty weight of 0.5.

COND-MERGE. Considers discards and merges, with a conditional probability metric for sharing penalty.

COND-ALL. Considers all moves, with a conditional probability metric for sharing penalty.

We define the initial value, or significance, of each input group g as $v_0(g) = min(|g|, sizeThreshold) \times msgCount(g)$. Intuitively, this captures the proportion of the corpus represented by g. The parameter $sizeThreshold$ prevents large groups, which are often once-off mailing lists, from being awarded excessively large initial values. Empirically, we set $sizeThreshold = 20$.

5.1 Algorithm Illustration

To provide insight into our algorithm, we first present its behavior on one user's data. As shown in Figure 4, all algorithm variants capture a significant fraction of the value with a small percentage of groups, with DISCARD, COND-MERGE, MERGE, and COND-ALL in increasing order of value captured for a given topology size. DISCARD allows no over-sharing; its sharing penalty is effectively ∞. COND-MERGE allows sharing mainly among those who are already sharing other messages. Next is MERGE, with a fixed penalty weight of 0.5, the algorithm is allowed to perform more merging.

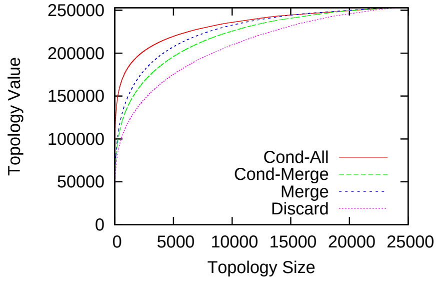


Figure 4: Social topologies for a representative data set.

COND-ALL has the highest compression ratio, though it actually discourages sharing in the final topology. Because the value of one group can be transferred to another with a sharing penalty, COND-ALL tends to identify the super individuals and groups that may play different roles in a user's interaction. Consider, for example, a secretary who is carbon-copied on all work-related emails. The secretary can amass a very large value as partial credit is transferred to him as low-frequency groups are dropped.

5.1.1 *Aggregate Behavior*

We glean additional insight about our algorithm's behavior by analyzing the frequency of the move types aggregated over the entire email data set. Figure 5 shows move frequency plotted against normalized algorithm progress.

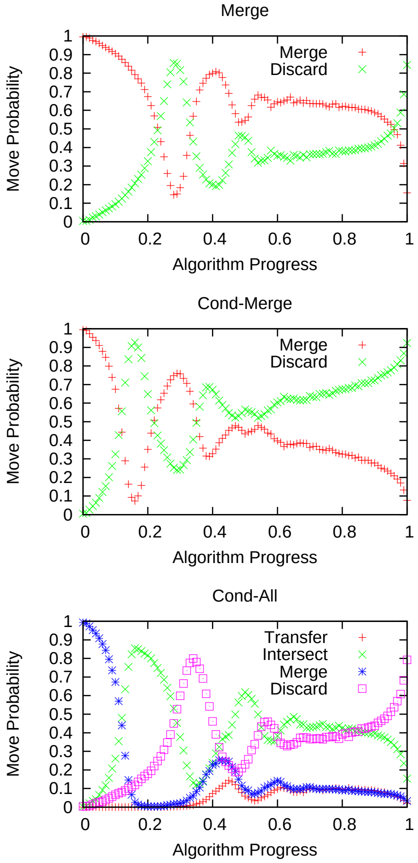


Figure 5: Algorithm behavior over the email corpus.

We see that in MERGE and COND-MERGE, there are distinctive alternating phases of merges and discards. The periodicity reduces over time with merges dominating at the beginning and discards dominating near the end. As the algorithm takes the move with the minimum value reduction, the periodicity results from the fact that there are many initial groups with values 1, 2, and so forth. Many discards of groups of value 1 kick in as the minimum drop in the algorithm reaches 1. Since the merged groups no longer have integral values, the choice between discards and merges become more irregular. Near the end of the algorithm, the remaining groups are distinct enough that merging them would incur a higher penalty than discarding them, thus we see many discards near the end. COND-MERGE is similar to MERGE, except that MERGE performs more unions since it has a lower sharing penalty.

COND-ALL has two more moves than MERGE: intersects and transfers. Almost all the intersect moves occur between supersets and subsets. In such cases, intersects produce the same topology as discards of the larger group, but the smaller group now accumulates more value due to the transfer of value. Including this move favors the creation of smaller groups and helps identify the core people in each group. Similarly, transfer moves also create pressure to produce smaller groups, since values can be transferred from one group to another. Together, intersect and transfer moves reduce the number of merges.

5.2 Value Concentration

Figure 6 plots the median fraction of summary groups that capture a given fraction of value.

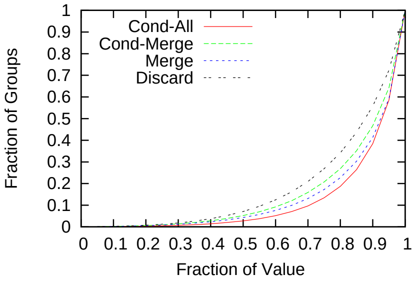


Figure 6: The values of social topologies obtained for the email corpus.

We see from the summary in Figure 7 that 50% of the value can be captured by 7% or less number of groups. If we are willing to tolerate some over-sharing, we can compress the social topology further. MERGE needs only 23% of the groups versus DISCARD's 34% to capture 80% of the value. COND-MERGE only supports merging of closely related friends, causing the need of slightly more groups. As discussed above, since COND-ALL allows the value of a group to be transferred to another, without having to include all members of the group, COND-ALL achieves the best value with the smallest number of groups. To reach 80%, COND-ALL needs a social topology whose size is less than 20% of the original.

	DISCARD	MERGE	COND-MERGE	COND-ALL
0.5	0.07	0.04	0.05	0.03
0.6	0.13	0.08	0.09	0.05
0.7	0.21	0.13	0.16	0.10
0.8	0.34	0.23	0.27	0.19

Figure 7: Fraction of groups needed to achieve a given fraction of the value.

5.3 Small Social Topologies

Since many users in our corpus have over 1,000 groups, even 10% of groups might be overwhelming for the user to review. How much value can be captured by a few tens of groups? Figure 8 shows the median of the values captured for the fixed size social topologies and Figure 9 tabulates the values for topologies with 10, 25, and 50 groups. We find that the top 10 groups capture 24-34% of the value and the top 50 groups capture 44-57%, depending on the algorithm variant used.

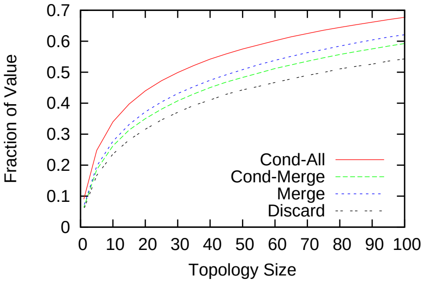


Figure 8: Values of small social topologies.

Our algorithm treats singletons the same as any other groups, allowing us to rank individuals uniformly against groups. However, certain applications may not need to be concerned with singletons. For example, a tool that helps users name groups only needs to show non-singleton groups, since individuals already have a name. We thus show the number of non-singletons in Figure 9 for reference. The majority of the top groups in email turn out, not surprisingly, to be singletons. As the allowance for over-sharing grows from COND-ALL, COND-MERGE, to MERGE, the number of non-singleton groups increases slightly. Thus for applications that work with only non-singletons, just 2-4 groups are needed to reach 24-34% of the value and 8-11 groups reach 35-47%.

	DISCARD	MERGE	COND-MERGE	COND-ALL
10	0.24 (2)	0.28 (4)	0.26 (3)	0.34 (3)
25	0.35 (8)	0.40 (11)	0.38 (10)	0.47 (8)
50	0.44 (21)	0.51 (25)	0.49 (24)	0.57 (18)

Figure 9: Values of social topologies with selected sizes. Non-singleton groups are shown in parentheses.

	DISCARD	MERGE	COND-MERGE	COND-ALL
Non-singleton	21	25	24	18
New groups	0	14	6	0
Group size	2.6	6.1	3.5	2.5
People	60	162	84	71
Roles/person	2.0	1.8	1.9	1.6

Figure 10: Properties of social topologies of size 50.

Which version of the algorithm should an application use? The different variants produce different topologies. Figure 10 shows additional properties of social topologies of size 50. It is clear from the figure that MERGE generates the largest social topology, followed by COND-MERGE. Perusal of the authors' own social topologies suggests that MERGE can create somewhat noisy groups consisting of people who are only peripherally related. Social topologies created by COND-MERGE are fairly coherent, and are the recommended choice for generating groups from social network data. On the other hand, COND-ALL is suitable for distilling key members of each group. We observe that no new groups are created for the COND-ALL case; since there is heavy traffic within the core groups, it is highly likely that there is at least one message sent to the entire core group.

5.4 Significant Groups

For applications without a fixed budget, it is useful to report to the user only the significant groups of his or her topology. Showing too many groups can bore the user, while showing too few might miss important groups. We can leverage our valuation framework to choose the appropriate number of groups to present. The average value of a group in the input data set serves as a baseline for the importance of a group. To identify groups that stand out above the average, we can simply display groups with value greater than one standard deviation above the average value of the input data. We can alter the algorithm to stop when the error for a move exceeds this threshold. As shown in Figure 11, the median of 11 groups was directly identifiable from the input data set (DISCARD); COND-ALL and COND-MERGE identify a median of 15 significant groups for the email data sets and MERGE identifies 14. These numbers appear to be quite reasonable in practice.

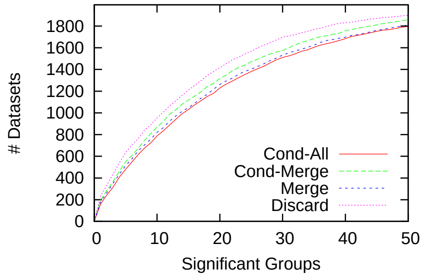


Figure 11: The cumulative distribution of the number of significant groups in the email corpus.

6. ANALYSIS OF PHOTOS

We analyzed the four variants of the greedy algorithm described in the previous section for Facebook photo tags. All plots shown represent the median observed in the data set.

6.1 Value Concentration

We observed the same overall trends with the photo data set as we saw in the previous section. In Figure 12, the fractional value curve climbs less steeply than in Figure 6, suggesting higher diversity in the photo data set compared to email. Figure 13 shows that COND-ALL requires only 15% of the groups to capture 50% of the value, and 42% to capture 80% of the value.

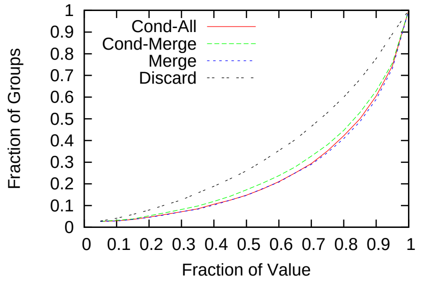


Figure 12: The values of social topologies obtained for the photo corpus.

	DISCARD	MERGE	COND-MERGE	COND-ALL
0.5	0.26	0.15	0.17	0.15
0.6	0.35	0.21	0.24	0.21
0.7	0.46	0.29	0.33	0.29
0.8	0.60	0.41	0.45	0.42

Figure 13: Fraction of groups needed to achieve given fraction of the value for photos.

One observed difference from email is that all variants other than DISCARD have almost identical curves. This suggests that the photo data set may be capturing tighter friendships since the trend of core friends tracked by the COND-ALL variant is similar to the MERGE variant which tends to create groups including more peripheral relationships.

From Figure 13, we see that DISCARD needs 26% of the groups to capture 50% of the value, whereas COND-ALL needs only 15%. That is, COND-ALL is better than DISCARD at compressing the social topology by a factor of 1.7. COND-ALL has compression improvement of 1.4 to 1.7 times for photos over DISCARD; it has an improvement of 1.8 to 2.6 for email.

6.2 Small Social Topologies

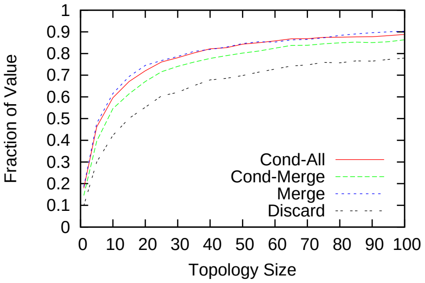


Figure 14: Values of small social topologies derived from photos.

If we wish to help users create Facebook friends lists to avoid over-sharing, it is important that we do not overwhelm them with too many groups. Even though a higher fraction of groups is needed than email, since photos are a smaller data set, the value is captured by a relatively small number of groups. Figure 14 shows the median of all values obtained for group sizes up to 100, and Figure 15 shows a few samples of the data. For example, with just 10 groups, 60% of value is captured by the COND-ALL algorithm, compared to 34% for the email data set. We see that the percentage of non-singleton groups is much higher, reflecting the fact that photo-taking is a gregarious activity, unlike email which often

	DISCARD	MERGE	COND-MERGE	COND-ALL
10	0.42 (8)	0.62 (9)	0.55 (8)	0.60 (7)
25	0.60 (21)	0.77 (21)	0.72 (21)	0.76 (18)
50	0.70 (42)	0.85 (41)	0.80 (42)	0.84 (37)

Figure 15: Values of photo-based social topologies with selected sizes. Non-singleton groups are shown in parentheses

involves correspondence with only one other person.

More characteristics of social topologies with 25 groups are shown in Figure 16. Note that the number of non-singleton groups included here are determined more by the data set than the algorithm. In this case, even the COND-ALL variant has a couple of new groups; it is harder to take a photo of a cohesive but broad group, whereas it is common to write at least one message to it. The median of the average group size is much higher across the board. MERGE still derives larger groups and includes more people, but not substantially more. The results show that people on average play about two roles, confirming the importance of our unique ability to find overlapping groups.

	DISCARD	MERGE	COND-MERGE	COND-ALL
Non-singletons	21	21	21	18
New groups	0	11	4	1
Group size	4.5	6.9	4.8	2.9
People	54	90	64	57
Roles/person	2.1	2.0	1.9	1.8

Figure 16: Properties of social topologies of size 25 from photos.

6.3 Significant Groups

Photo tagging data shows a distinctly lower number of significant groups than the email data set. In Figure 17 we see that the photo data set has a median of 7 significant groups.

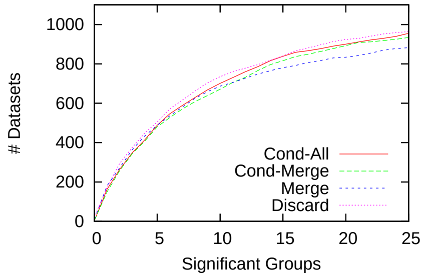


Figure 17: The cumulative distribution of the number of significant groups for the photo corpus.

7. EVALUATION BY EDIT DISTANCE

We now compare our algorithm variants with Newman's fast greedy clustering algorithm [17], which is a commonly used algorithm for discovering communities in social graphs. For this purpose, we used the implementation of Newman's algorithm in the igraph package of R. Unlike our algorithm, Newman's algorithm partitions the nodes in the graph into clusters via optimization of a modularity metric. As a neutral objective function, we select *edit distance*, an information-theoretic metric that is not a direct objective for either our algorithm or Newman's.

The *edit distance* between two words is defined as the minimum number of character alterations required to modify one of the words until it is equivalent to the second. We employ a modified version of edit distance for group communication data. The edit distance for a collection of communications C given a social topology S is

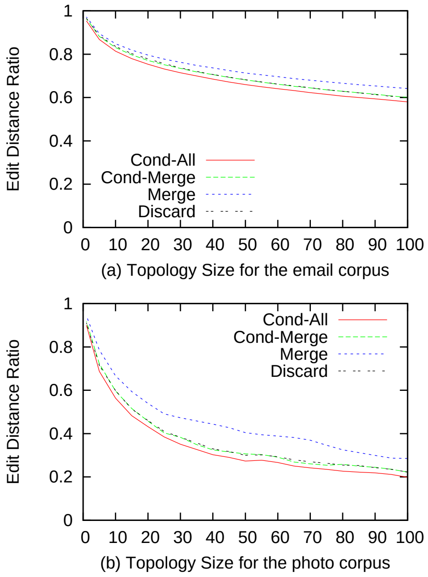
$$\text{EditDistance}(S, C) = \sum_{c \in C} \min_{s \in S} |c \cup s| - |c \cap s|$$

Intuitively, this metric captures the minimum number of insertions and deletions needed to specify the participants for each communication given a topology. The edit distance for each group is the number of members added and subtracted from its closest matching group in the topology. The sum of such edits defines the edit distance of a topology with respect to a set of groups. The largest possible edit distance is simply the sum of the sizes of the input groups.

We performed an experiment where we compute the edit distances for both the email and photo data sets using Newman's algorithm and the four variants of our algorithm. Each sent message and each tagged photo is treated as one unit of communication. We simply treat all the clusters generated by Newman's algorithm as the social topology for a user. As shown in Figure 18, the ratios of the minimum edit distance to the maximum edit distance (the total size of all the input groups) are similar for the two types of data, with the medians being 0.93 and 0.84 for email and photos, respectively.

	Group Size	# Groups	Edit Distance Ratio
Email	1	118	0.93
Photos	3	6	0.84

Figure 18: Median group parameters and edit distance ratios for Newman clustering.



(a) Topology Size for the email corpus

(b) Topology Size for the photo corpus

Figure 19: Comparison of the EditDistance metric across all 4 algorithm variants for the (a) email corpus and (b) photo corpus.

For our social topology algorithm, edit-distance ratios obtained is a function of the number of groups in the social topology. The medians of the edit-distance ratios are thus plotted in Figure 19. The results show that our algorithm outperforms Newman clustering in minimizing edit-distance ratios. All variants of our algorithm beat the clustering algorithm with just 4 groups for email and 3 groups for photos. There is a significant difference in edit-distance ratios between the email and photo datasets. 10 groups generated by COND-ALL yield median ratios of 0.81 and 0.56 for emails and photos respectively; 25 groups yield ratios of 0.74 and 0.38.

Edit-distance ratios do not differ significantly between social topology algorithm variants, but we note that MERGE produces a worse edit-distance ratio than DISCARD. Given that MERGE uses a penalty weight of 0.5 for over-sharing whereas the penalty of a deletion for edit distances is 1, this makes sense. The goal of MERGE is find related people and not to optimize edit distance. Similarly, it is not expected for Newman's algorithm to produce small edit-distance ratios either. We performed this comparison mainly to illustrate how our algorithm is different from standard clustering algorithms. Our algorithm aims to identify the significant, possibly overlapping, groups where individuals may play multiple roles.

8. CONCLUSION

Unlike most other social network analysis algorithms that detect groups from global network data, our algorithm helps individuals automatically identify and use their social groups by analyzing their online social actions.

We formulated the social topology extraction problem as the compression of a natural social topology, where initial groups are labeled with their significance value, to a desired size according to a metric function that biases the composition of desired groups. We proposed a simple greedy algorithm derived from this value metric. Our algorithm can be used to produce the best representation of a social topology for a given size budget, though it can also automatically determine the number of significant groups a user has.

We have made publicly available two applications based on our algorithm to help users define friends groups and lists based on email and photo tags[3]. We are encouraged by the enthusiasm expressed by our users; the applications have been well received and it appears that the results are good enough to be interesting to many users. Our algorithm and source code are publicly available, and can be downloaded at the above URL.

We have performed an analysis of our algorithm over approximately 2,000 email archives and 1,100 photo collections, the latter collected by our Facebook application. We show that our algorithm is significantly different from the popular Newman's clustering algorithm for community detection. Using edit distances as an information-theoretic metric, we see that even a tiny topology consisting of 4 groups for email and 3 groups for Facebook produces significantly smaller edit distance ratios than Newman's algorithm. Our algorithm, with its ability to find nested and overlapping sets, is designed to find significant groups of friends from social data.

We found that both the email and photo corpus are highly amenable to compression, allowing our algorithm to produce social topologies that capture much of the value in the input set with a small percentage of groups. We show that the algorithm can capture 80% of the value with 20% and 42% of the groups for email and photos, respectively. More excitingly, we found that there are less than 15 significant groups in our email communications and 7 groups in photos for half of the population in our experiment. The results demonstrate the ability of our algorithm to distill out a small number of groups from thousands of emails and hundreds of photos. It also offers insight into people's social relationships as captured by their online activities.

[3] http://mobisocial.stanford.edu/groupgenie

9. ACKNOWLEDGMENTS

The authors would like to thank Peter Monaco from Xobni for providing the anonymized email data set. This research is supported in part by the NSF POMI (Programmable Open Mobile Internet) 2020 Expedition Grant 0832820, NSF grant CCF-0964173, Stanford Clean Slate Program, and Stanford MobiSocial Computing Laboratory.

10. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. Mooney. Model-Based Overlapping Clustering. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 532–537, 2005.

[3] A. Clauset, M. Newman, and C. Moore. Finding Community Structure in Very Large Networks. *Physical Review E*, 70(6):66111, 2004.

[4] A. Culotta, R. Bekkerman, and A. Mccallum. Extracting Social Networks and Contact Information from Email and the Web. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[5] Facebook groups: How do I chat with a group? http://www.facebook.com/help/?faq=18808.

[6] Facebook featured friends: How do I feature specific friends on my profile? http://www.facebook.com/help/?faq=19417.

[7] E. Gilbert and K. Karahalios. Predicting Tie Strength with Social Media. In *CHI '09 Proceedings of the 27th International Conference on Human Factors in Computer Systems*, pages 211–220, 2009.

[8] J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proceedings of the IEEE Symposium on Information Visualisation (InfoVis 2005)*, pages 33–40, 2005.

[9] G. Kossinets and D. Watts. Empirical Analysis of an Evolving Social Network. *Science*, 311(5757):88, 2006.

[10] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the Overlapping and Hierarchical Community Structure in Complex Networks. *New Journal of Physics*, 11:033015, 2009.

[11] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical Properties of Large Social and Information Networks. In *In Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pages 695–704, 2008.

[12] LinkedIn maps. http://inmaps.linkedinlabs.com/.

[13] D. Maclean, S. Hangal, S. K. Teh, M. S. Lam, and J. Heer. Groups Without Tears : Mining Social Topologies from Email. In *Proceedings of the 2011 International Conference on Intelligent User Interfaces*, pages 83–92, 2011.

[14] 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.

[15] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph Summarization with Bounded Error. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 419–432, 2008.

[16] S. Navlakha, M. Schatz, and C. Kingsford. Revealing Biological Modules via Graph Summarization. *Journal of Computational Biology*, 16(2):253–264, 2009.

[17] M. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69:066133:1–5, 2004.

[18] 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.

[19] 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 *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 233–242, 2010.

[20] M. G. Siegler. Zuckerberg: "Guess What? Nobody Wants To Make Lists", August 2010. Retrieved from: http://techcrunch.com/2010/08/26/facebook-friend-lists/.

[21] T. Simonite. Facebook app reveals your social cliques, February 2011. http://www.technologyreview.com/communications/32394.

[22] Y. Tian, R. Hankins, and J. Patel. Efficient Aggregation for Graph Summarization. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 567–580, 2008.

[23] 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.

[24] S. Whittaker, Q. Jones, B. A. Nardi, M. Creech, L. Terveen, E. Isaacs, and J. Hainsworth. ContactMap: Organizing Communication in a Social Desktop. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(4):445–471, 2004.