

Towards Automatic Recommendation of Friend Lists

Kelli Bacon
Computer Science
University of North Carolina
Chapel Hill, USA

Prasun Dewan
Computer Science
University of North Carolina
Chapel Hill, USA

Abstract— Facebook access-control lists, called friend lists, are difficult to create manually. Previous work on automating these lists has used friend details entered by the user. Our approach automates them by merging virtual friend cliques using certain heuristics that determine if two virtual friend cliques correspond to a single actual friend clique. A small user study found that our lists were more likely to be used by participants than existing automated lists.

Keywords- *Social network; clique; recommender system; friend list; Facebook; access control; mixed-initiative; social mining*

I. INTRODUCTION

Social networking is a growing area in collaborative computing. Especially popular with students, some studies show that they spend as much time social networking as they do studying [11]. The ever-growing, ever-evolving Facebook is an important state-of-the-art example of a system supporting social networking.

One of the reasons for the growth of Facebook is the notion of a friend list, which is a personally-named list of people one has declared as friends. Facebook was originally only open to students at select universities. As it added features to its interface and the user-base grew, it introduced the limited profile friend list. This list provided users the opportunity to restrict parts of their profiles to people of whom it would be socially unacceptable to deny a “friend request” but with whom users did not want to share everything.

Thus a friend list was a special form of the traditional notion of an access control list, where the protected objects were not data owned by users but information about the users. Today, Facebook users can create and manage multiple friend lists that provide a variety of other forms of privacy controls. For instance, they protect the various kinds of profile information seen by users and who can chat with them. In fact, they serve not only as personal access control lists but also as mailing lists to send messages to a subset of a person’s Facebook friends. Then in early July 2009, Facebook announced that it was working toward allowing individual posts (from status messages to shared links) to be directed at friend lists [10], which seems like a timely feature given news stories of people losing their jobs because of status posts [5]. It would provide users with an opportunity to, for example, post work-related information to coworkers while keeping reunion pictures to the high-school-buddy list.

The research on SOYLENT [2] suggests another use for user groups such as friend lists—they can help users explicitly identify and understand the social structures to which they belong. It might also be useful to make these structures visible to others as automatically generated public friend details. Thus friend lists can potentially provide an extra layer of classification for a social network that can be used for access control, email, and understanding of social structures. Perhaps for these reasons, the designers of Facebook now consider them so fundamental that when users add new friends, they are prompted to add them to one or more friend lists.

However, a relatively few number of Facebook users understand and use the notion of friend lists [11]. Moreover, our personal experience and the results of a small survey reported here shows that it takes time and effort to organize friend lists, which many users are not willing to put forth. This is consistent with the high effort of creating access control lists in general [9]. Recently, recognizing these two problems, Facebook automatically created friend lists for their users based on friend details explicitly entered by users when they formed friendships. Friends sharing some friend detail were put in an automatically created friend list named by the common friend detail. For example, all friends whose friend details indicate they are family are put in a friend list called family.

The idea of automating the creation of friend lists has the potential for tremendously increasing the use of friend lists in particular and user groups in general. The work done by Facebook is an important step in this direction. For example, the automated family list generated for the second author included several of his family members.

However, this work has several problems. First, the free-form nature of friend details can result in multiple friend lists for the same social group. One problem the first author experienced was the appearance of three automatic friend lists for the same student-government group (GSBA, GSBA-Senate, and Senate). Second, and more important, the Facebook approach can group only those friends who have joined networks or for whom friend details have been filled out. Our experience shows that such information is not available for a large number of users. Finally, no research has been published on the validity of the automated friend lists.

This work takes an initial step in addressing these limitations.

II. IMPLICIT GROUPS FROM FRIEND CLIQUES

A. Recommendations based on implicit information

The work on SOYLENT has suggested that it is possible to overcome the second problem of the Facebook approach by grouping users based on implicit information about them kept by the system that defines these users. Specifically, the email conversations of users provide a basis for grouping them (to different degrees). It is possible to extend this approach to a social network by looking at not only asynchronous messages but also other interactions among users in the network, such as Facebook wall postings, synchronous chat messages, and comments on various kinds of postings.

However, this approach has the following problem when applied to the specific case of Facebook – the system used in our study – because Facebook applications are not supplied this information. Although much of this information about users is “publically” available¹ to other users, Facebook policies prohibit the collection of this information by web crawlers. Thus, at least in the short term, this approach may not be implementable by software that is not deployed by Facebook.

One piece of information that is available to Facebook applications is the public friendship relationships among the friends of the users running the applications. The hypothesis of this research is that this information can be used to recommend, but not fully automate, friend lists to users.

The reason we do not strive for full automation is that no system can perfectly guess the friend lists desired by all users no matter how much information it has about the user. An important reason for this is that different users may use these lists for different purposes. A system that groups users based solely on friendship relationships is bound to create lists that are far from absolute. Our approach is intended to recommend friend lists to users which can then be manually morphed or completely rejected by them. This is consistent with the goal of mixed-initiative computing in general and mixed-initiative access control in particular [1]. It is also consistent with the idea in TagAssist [12] wherein the guessed tags are not applied automatically but suggested for the user to choose from.

B. Actual and Virtual Friend Cliques

Our approach is centered on the notion of friend cliques. In a social context a *clique* is a tightly-knit group of people. In graph theory, it has a more formal definition: a clique is a set of vertices in a graph that are all connected to each other by edges of the graph. By this definition, every subset of a clique is also a clique. A *maximal clique* is a clique that is not a subset of a larger clique.

Graph cliques are *human cliques* when the vertices of a graph represent users. When the edges of such a graph represent friendship relationships, the human cliques are *friend cliques*. Depending on whether these are actual or virtual

friendship relationships, the cliques represent *actual* or *virtual friend cliques*. In the rest of the paper, we will use the term clique to refer to a maximal friend clique.

Considering friendship relationships among the friends of the user for whom the friend lists are computed, our definition of cliques takes a *network-centric* view of social groups. This is in contrast to the approach taken in SOYLENT of taking a user or *ego-centric* view, where the relations between the contacts of the user for whom the groups are identified are not considered. In SOYLENT, if user A sends the same message to users B, C, and D, then B, C, and D are grouped together regardless of whether they send messages to each other. Taking an ego-centric view in our project would result in a single friend list as we do not have information about the activities of a user.

It has been observed that actual cliques form meaningful technology-supported social groups, such as a group that may wish to watch TV together despite being geographically separated [7]. There are other social groups that remain socially close (without regard to geographic location) via social networking sites like Facebook. The main ideas in our approach are that (a) an actual clique forms a group that should be recommended as a friend list to each member of the clique; and (b) actual cliques can be approximated from virtual cliques.

The major problem with this approach is that actual and virtual cliques can be very different. The reason is that an online social network can mimic the relationships or connections people experience in the real world, but is rarely a perfect representation of the real social network. There are many instances in which two users belong to the same social group, but are not connected by a social network for any number of reasons. For instance, one of the users may be new to the networking site and, therefore, may not have made all the connections. Similarly, the two users may be acquaintances but not “friends. A symptom of this problem is the large number of virtual cliques into which a user’s friends can be grouped. For example, a participant in our study with less than 130 friends had as many as 3,021 virtual cliques! Moon and Moser [6] show that a graph with n nodes will have at most $3 \sqrt{3}$ maximal cliques. Fortunately our participants did not have the maximal number of maximal cliques, but the number of maximal cliques we did observe was still very high, ranging from 61 to 3,915. It seems safe to assume that the number of actual cliques to which they belong would be much smaller.

C. Merging Virtual Friend Cliques

Deriving actual cliques from virtual cliques requires identification of the missing links. These missing links result in actual cliques being split into multiple virtual cliques, leading to the clique explosion problem. Given the set of virtual cliques, we must combine virtual cliques with a high degree of commonality to approximate a smaller set of actual cliques. To illustrate this approach, consider the social network in Figure 1 where the edges represent virtual friend relationships. It contains three maximal cliques:

¹ Krishnamurthy et. al. encountered a similar accessibility issue of getting data from a system; they could collect “following” but not “followed by” on Twitter [3].

This research was funded in part by NSF grants IIS 0312328, IIS 0712794, and IIS 0810861.

- (1) Kami, Betty, Dylan, Gene
- (2) Betty, Dylan, Gene, Elle
- (3) Taj, Elle

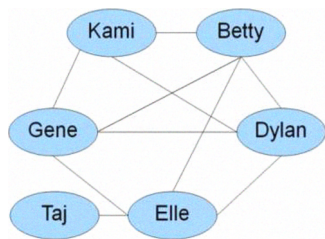


Figure 1. An example of a social network

These three cliques are a subset of actual virtual cliques of one of our participants - the names have been changed to preserve anonymity. In this example, Kami and Elle do know each other, but are not friends on the social network. Thus, virtual cliques (1) and (2) should be merged with each other.

Clique merging is not a new problem in graph theory. Li, Tan, Foo, and Ng have constructed an algorithm that merges maximal cliques for the purpose of predicting protein complexes [4]. Our goal was to find an analogous algorithm for social networks.

III. BUILDING A FRIEND LIST RECOMMENDATION

We acquired mutual friendship data using a Facebook application. The mutual friendship data allowed us to create a social network. We used the Facebook unique identifiers in order to avoid a situation in which a user had two friends with the same name. This mutual friendship data was entered into a JGraphT [8] simple undirected graph. The social network of each participant was represented as a simple undirected graph because in Facebook, friendships are reflexive and only one connection exists between each pair of users.

The following pseudo code describes our merging approach formally.

```

cliques <- getAllMaximalCliques2
loop
  merged = false;
  for each cliqueA in cliques
    for each cliqueB in cliques
      merged = mergeCliques(
        largerClique, smallerClique)
  if (!merged) exit;
end

```

The algorithm continues to merge cliques until no more merges are possible. There are several heuristics possible to determine when two virtual cliques have enough commonality to warrant merging them. We expect that the exact set of heuristics to be used is a matter of much future research, especially for different clique merging applications. To determine if the merging approach for social networks has promise, we did some top-down thinking and bottom-up experiments on training data to identify two initial algorithms. The threshold numbers 0.15, 0.9 and 0.35, given in the algorithms below, are the results of these experiments.

Our first algorithm performs merges on two cliques based on the number of items in the smaller set that are not in the larger set normalized by the size of the smaller set.

```

mergeCliquesSetDiff(A, B)
  BMinusA = B - A
  percentDiff = |BMinusA|/|B|
  if percentDiff ≤ .15 //fifteen percent
    A = A + BMinusA
    return true;
  return false;

```

This algorithm computes only the differences between the two sets. Our second algorithm considers directly both similarities and differences, determined by computing set intersections and differences, respectively.

```

mergeCliquesIntersection(A, B)
  AintersectB = A ∩ B
  BMinusA = B - A
  percentSame = |AintersectB| /|B|
  percentDiff = |BMinusA|/|B|
  if percentSame ≥ .9 || percentDiff ≤ .35
    A = A + BMinusA
    return true;
  return false;

```

. These algorithms were implemented in Eclipse using JDK compiler 1.6. In the case of both algorithms, cliques with fewer than three members were not recommended as friend lists. We also experimented with several variations of these algorithms. These variations include (a) computing $A - B$ instead of $B - A$, (b) normalizing by the size of A (the larger clique) instead of the size of B , (c) using different thresholds in the two algorithms, and (d) performing only one merge pass instead of repeatedly merging until no more merges are possible. However, each of these variations gave inferior results with the training data.

IV. EVALUATION

A. Pilot Study

The initial study consisted of experimentation with the social networks of five Facebook users, which included the first author. Each participant was a college student or college graduate who joined Facebook while in college. For the best variations of the mergeCliquesSetDiff method on the training data, we sent the resulting friend lists to the participants and asked them to return to us the number of *good lists* - lists for which a unique description could be provided that applied to the vast majority of the friends in the list. The accuracy of a method was the ratio of this number to the total number of lists.

There are several other reasonable definitions of accuracy possible. In our first-cut at defining this new concept, we were driven by two concerns: (1) participant evaluation effort should be low; (2) participant privacy should be respected. Goal (1) prevented us from asking each participant to count the number of missing and invalid items in each list. As a result, we could not quantify the usefulness of each list. Goal (2) prevented us from requiring the descriptions of the good lists, as friend lists (and their names) are private in Facebook. As a result, we could not determine the kind of good lists we found such as

² Computed using the BronKerboschCliqueFinder in JGraphT This function returns all of the maximal cliques as a collection of sets of the vertex values.

family, work, or church. Our evaluation approach is consistent with the approach taken in SOYLENT of evaluating its visualizations of user-groups by showing them to each user and asking if they are meaningful, letting “meaningful” being defined by each user.

B. Further Investigation

As the first study provided insight into which threshold numbers to use, we conducted a larger follow up study involving ten participants with both of the algorithms. None of the authors was a participant in this study. This time, we asked the subjects to not only tell us the number of good lists but also answer a questionnaire that asked them to (a) indicate how familiar they were with friend lists, how useful they considered the generated lists, whether they had set up friend lists, and what kind of friend lists they had set or would set; and (b) qualitatively compare the two sets of friend lists generated by our algorithms and the one generated by Facebook.

C. Results

Both algorithms had accuracies of about 80%, as seen in Table 1. Again, accuracy was measured as the number of “good lists” over the number of lists that had been recommended. This is a strong result given that some subjects had thousands of virtual cliques – modulo limitations of our accuracy metric, our algorithms essentially found needles in large haystacks with information only about friendship relationships. We also considered “coverage” in our evaluation; a 100% accuracy rating is not very impressive if it reflects only one recommended list covering 10% of the participants’ friends. Thus “coverage” measures the number of friends that are in the union of all friend lists divided by the total number of friends. We are not concerned if a friend appears in more than one list as it is common for some social groups to overlap (ex: theater club and choir).

Overall, participants preferred the recommendations by mergeCliquesSetDiff despite it having a slightly lower accuracy and a significantly lower coverage. They overwhelmingly preferred these lists to those generated by Facebook. One respondent said “the recommended lists are better because they are much more specific and can isolate specific groups within a given network.” Another said that the recommended lists were “more real” than the automatically generated ones.

A couple of the participants supplied the number of lists Facebook had automated for them as well as the number of those lists which were unique, or “good”, and a description of how accurate these lists were. Most of these participants noted that although the Facebook lists had accurate descriptions (which is to be expected as these are based on friend details explicitly entered), those lists were incomplete to the extent of being useless. The other participant claims that “the names on the [Facebook automated] lists are incorrect in their associations and intermingled.” In this comment she was referring to “College” list that Facebook created for her, which included persons from her undergraduate and graduate institutions.

As anticipated, most of the subjects (before this study) were not familiar with friend lists or Facebook’s attempt to automate

TABLE 1. ACCURACY AND COVERAGE RATING

Participant	Intersection Merge		Set Difference Merge	
	% “good”	% coverage	% “good”	% coverage
1	100	93.5	100	40.6
3	71.4	85.7	100	65.1
5	66.6	92.2	75	50.9
6	87.5	91.3	40	39.9
7	100	79.8	50	63.1
9	85.7	86.7	86	78.9
10	60	87.9	100	59.1
average	81.6	89.5	79	56.8

them. A majority did not set up friend lists because they believed it took too much time. Many of the participants considered friend lists to be useful for privacy and messaging purposes. One specifically said he would use friend lists to control who could see him in Facebook Chat. Two of the participants said they could be helpful for sending invitations to events. One of the participants said that “if the friend lists could somehow be imported into my Gmail with corresponding e-mail addresses I would totally use friend lists. I think the utility of something like that is pretty clear.” His comment is consistent with the findings of [2], and shows that that there is a relationship between friend lists found by merging cliques and mailing lists identified by examining interaction histories. On the other hand, one of the participants said she would have no use for the good lists we generated. The responses might have been different had we given the participants a menu of potential uses for these lists, which is still a matter of research given the evolving applications of them in Facebook.

Most participants noted that they would likely organize their friends by common school, work, or family, which is consistent with the responses collected by [11]. A few participants wished to classify groups further by clubs or activities the friends had in common. Specifically, participant 9 said she would organize her friend “by place I know them, i.e. UNC, Lehigh, lacrosse, Ursinus, etc.” One participant would prefer to have friend lists based on the interaction with friends on Facebook while another would group friends by the common language spoken.

Because of the privacy concerns mentioned earlier, we could not require participants to tell us what kind of social groups were actually captured by the good lists. However, we can give information from a participant who supplied the names of groups: participant 9 (mentioned above). She reported seven distinct groups in the results of the Intersection Merge, “Ursinus, Florida Trip, UNC/NC, High School, NC Lacrosse Girls, Lehigh Friends, Lehigh CS”, though “1 group [was] split into 2 subgroups”. Thus, this participant would have further merged two of the groups we recommended to her. When this participant looked at the recommendations from the mergeCliquesSetDiff algorithm, she found the following groups: “Ursinus Lacrosse, UNC CS Only, NC Lacrosse, Lehigh Friends, Ursinus Greek, UNC, High School.” She was

pleasantly surprised by some of the groups. "I like the separation of my Ursinus friends into lacrosse and Greek life though, since that is how the social groups really were." However, she would have merged the UNC groups in this case.

The above example suggests that a recommender system for friend lists should provide a scheme for not only naming and editing recommended lists, but also merging them. Assuming such an operation is provided, it seems better to err on the side of recommending small lists (such as UNC CS and UNC in the case of participant 9-Set Difference) than large lists (such as Ursinus in the case of participant 9-Intersection), as merging a group of sets requires a single operation, while splitting a set of size N into subsets requires $O(N)$ operations.

V. PRELIMINARY CONCLUSIONS AND FUTURE WORK

Based on our small study with ten participants, this paper makes several contributions, which include, (a) identification of the limitations of manual friend lists, (b) two initial metrics for evaluating recommended friend lists, (c) showing that actual cliques are split into a large number of virtual cliques because of missing virtual friendship relations, (d) two network-centric merging algorithms for reconstructing actual cliques from virtual cliques, and (e) an evaluation showing that the reconstructed virtual cliques are preferred as friend lists to those generated using friend details and have high accuracy.

Our evaluation does not distinguish between the two merging algorithms. However, the experience of the two authors and the participants shows that `mergeCliquesIntersection` tends to find large networks, while `mergeCliquesSetDiff` is more likely to find subgroups (such as club membership lists) within a larger network (school). It would be useful to improve the evaluation method to bring this out and develop a single algorithm that finds both networks and subgroups. It would also be useful to integrate our algorithm with the work on SOYLENT by mapping social network identifiers to email addresses, when possible. More important, it would be useful to quantitatively compare the effort required to morph recommended friend lists with those creating them from scratch. It would also be useful to do a larger study and separately evaluate the various uses of the recommended friend lists such as generating friend details and access control.

The overall contribution of the paper is to show the usefulness of pursuing these research directions.

ACKNOWLEDGMENT

Prof Wei Wang's Spring '09 class on data mining provided some of the background needed for this research.

REFERENCES

- [1] Dewan, P., Grudin, J., Horvitz, E. "Toward mixed-initiative access control." Colcom, pp.64-71, 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom) 2007.
- [2] Fisher, D. and Dourish, P. 2004. "Social and temporal structures in everyday collaboration," ACM CHI, pp.551-558, 2004.
- [3] Krishnamurthy, B., Gill, P., and Arlitt, M. "A few chirps about twitter," ACM WOS, pp.19-24, 2008.
- [4] Li, X., Tan, S., Foo, C., Ng, S. "Interaction graph mining for protein complexes using local clique merging," *Genome Inform*, 16(2):260-269, 2005.
- [5] Lo, P.K., "Fired Over Facebook Status." NBC San Diego. 3 Mar 2009. <http://www.nbcsandiego.com/around-town/archive/Fired-Over-Facebook-Status.html>
- [6] Moon, J., Moser, L, "On cliques in graphs," *Isr. J. Math.*, vol. 3, pp. 23-28, 1965
- [7] Nathan, M., Harrison, C., Yarosh, S., Terveen, L., Stead, L., and Amento, B. "CollaboraTV: making television viewing social again," *ACM UXTV*, vol. 291, pp.85-94, 2008.
- [8] Naveh, B., Jgrapht. <http://jgrapht.sourceforge.net>
- [9] Reeder, R. W., Bauer, L., Cranor, L. F., Reiter, M. K., Bacon, K., How, K., and Strong, H. "Expandable grids for visualizing and authoring computer security policies," *ACM CHI*, pp.1473-1482, 2008.
- [10] San Miguel, R., "Wallflower or Extrovert? Facebook Lets Users Choose by Post." *TechNewsWorld*. 2 Jul 2009. <http://www.technewsworld.com/story/67499.html?wlc=1246572325>
- [11] Skeels, M. M. and Grudin, J. 2009. "When social networks cross boundaries: a case study of workplace use of facebook and linkedin," *ACM GROUP*, pp.95-104, 2009.
- [12] Sood, S., Owsley, S., Hammond, K., Bimbaum, L. "TagAssist: Automatic Tag Suggestion for Blog Posts," *International Conference on Weblogs and Social Media*, 2007.