

# Friend Grouping Algorithms for Online Social Networks: preference, bias, and implications

Motahhare Eslami, Amirhossein Aleyasen, Roshanak Zilouchian Moghaddam, and  
Karrie Karahalios

University of Illinois at Urbana-Champaign, Computer Science Department,  
Urbana, IL, US

{eslamim2, aleyase2, rzilouc2, kkarahal}@illinois.edu

**Abstract.** Managing friendship relationships in social media is challenging due to the growing number of people in online social networks (OSNs). To deal with this challenge, OSNs' users may rely on manually grouping friends with personally meaningful labels. However, manual grouping can become burdensome when users have to create multiple groups for various purposes such as privacy control, selective sharing, and filtering of content. More recently, recommendation-based grouping tools such as Facebook smart lists have been proposed to address this concern. In these tools, users must verify every single friend suggestion. This can hinder users' adoption when creating large content sharing groups. In this paper, we proposed an automated friend grouping tool that applies three clustering algorithms on a Facebook friendship network to create groups of friends. Our goal was to uncover which algorithms were better suited for social network groupings and how these algorithms could be integrated into a grouping interface. In a series of semi-structured interviews, we asked people to evaluate and modify the groupings created by each algorithm in our interface. We observed an overwhelming consensus among the participants in preferring this automated grouping approach to existing recommendation-based techniques such as Facebook smart lists. We also discovered that the automation created a significant bias in the final modified groups. Finally, we found that existing group scoring metrics do not translate well to OSN groupings—new metrics are needed. Based on these findings, we conclude with several design recommendations to improve automated friend grouping approaches in OSNs.

**Keywords:** Automated Grouping, Clustering Algorithms, Online Social Networks

## 1 Introduction

Mailing lists, chat groups, Facebook lists, and Google+ circles are a few examples of tools that facilitate group creation in social media. We create groups to help us manage large amounts of information, in this case people. By creating a mailing list for an alumni group, we no longer need to memorize a long list of names. Instead, we can recall the group name and use it for exchanging messages [9]. In the context of OSNs, in 2007, Facebook introduced friend lists, manually created lists of Facebook friends, for the purpose of selectively sharing and reading content [24]. Twitter introduced lists

in late 2009 for filtering content from one's network [25]. In 2011, Google+ introduced circles that enable selective sharing and filtering of posts on the site. Recent studies have emphasized the desire and feasibility of grouping for privacy control, sharing, and filtering [14, 16, 15, 27]. These studies found that people desired groupings or clusters of members in their community. However, due to the high cost of creating groups manually, the majority of manual group creation mechanisms remained underused. A case in point was the 2010 Facebook announcement that only about 5% of Facebook users had created at least one Facebook list [7].

Given the significant burden of manual grouping, later work in OSN group creation proposed automating group creation while allowing users to modify the created groups [14, 16]. Following this philosophy, in 2011, Facebook introduced smart lists. Smart lists differ from the original Facebook lists in that they use a recommender system to automatically assign friends to different groups. Example groups include close friends, acquaintances, family, and others [23]. Similarly, recommendation-based tools such as FeedMe [4] and ReGroup [2] suggest recipients for a post based on prior sharing patterns and the content. Such automated recommendation-based techniques can be helpful in social media systems such as email. However, when applied to large, public OSNs such as Facebook, Google+ and Twitter, these techniques put a relatively high burden on users to verify friend suggestions—for every contact individually. If one user sends ten messages on an OSN, this requires verifying all of the recipients for all ten messages.

But automating group creation and allowing user modification need not to be limited to recommender systems. One can utilize clustering algorithms to create populated groups from the onset, and then allow for personal curation. While the feasibility of structural network clustering for group creation in social networks has been investigated before [14], less is known about the benefits and drawbacks of using various automated clustering algorithms for grouping people within a social media interface. This work is a first step in that direction.

In this paper, we present a grouping tool that automatically creates groups within Facebook using three different clustering algorithms: Markov Clustering, OSLOM, and Louvain. The interface then enables the users to modify the groupings as needed. To verify the usefulness of our tool and to compare the effectiveness of the three algorithms, we evaluated our tool using both human perception and traditional clustering evaluation metrics. The following summarizes our three major findings:

- We found that users preferred automated groupings with the proposed graphical tool over existing manual or recommendation-based grouping tools such as Facebook smart lists. In addition, two of the three clustering algorithms we evaluated (Markov Clustering and Louvain), performed significantly well in terms of human satisfaction and traditional clustering evaluation metrics. These algorithms are appropriate candidates for automated friend grouping applications.
- Comparing the final groupings from different algorithms created by each participant, we found a significant difference between these groupings (14%). This relatively high difference illustrates a bias resulting from the automation in users' final groupings. We argue this bias arises primarily from (1) being influenced by the al-

- gorithmic groupings, (2) the existing hierarchical structure in social relationships, (3) having friends with multiple roles and (4) the user’s uncertainty when grouping.
- We explored group composition before and after modification based on two validated and efficient metrics that assess the quality of groups in the absence of ground truth: *Conductance* and *Triad-Participation Ratio (TPR)*. We found four categories of groups that did not fit the traditional definition of a group assumed by these metrics. We posit that such groups which exist in social network sites such as Facebook, therefore, require different group quality assessment metrics.

In the following section, we begin by reviewing previous studies on friend grouping in OSNs. Then, we introduce group detection in networks and the three clustering algorithms we used to build our automated friend grouping tool. After explaining our mixed-methods study, we discuss the results of our study using both quantitative and qualitative evaluations. We conclude by suggesting future directions for friend grouping algorithms and interfaces.

## 2 Literature Review

Selective sharing, filtering of content and privacy control are cited as major motivators for the creation of groups on OSNs. Early work exploring group creation focused primarily on privacy control interfaces [14, 16]. In this domain, manual creation and annotation of groups was costly in terms of time and frustration due to unintuitive interfaces. This approach resulted in a lack of use of personalized, curated privacy settings [13]. While these studies emphasize privacy, the implications extend to information filtering and selective sharing [15]. Studies on group creation demonstrate that people are not willing to use current grouping techniques in OSNs as they were intended. For example, a study on Facebook lists at 2010 showed that only 20% of participants’ friends were included in Facebook lists and none of the participants used these lists for controlling privacy [14]. In a related study, Kelly et al. asserted that participants using Facebook lists to create groups included few friends [16]. A 2012 study of Google+ notes that although users perceive grouping friends on OSNs positively, Google+ circles were only moderately used to selectively post to groups and filter incoming content [27].

Jones and O’Neill’s suggest that existing list and grouping tools have not met expectations [14]. They conducted a study asking people to create groups of their Facebook friends to apply group privacy settings. They discovered that organizing contacts into groups required too much time and effort; therefore, users were unwilling to group in this manner. Similarly, Kelley et al. [16] conducted a study asking users to apply four manual strategies (card sorting, grid tagging, file hierarchy and Facebook friend lists) to create groups in Facebook. They suggested that assistance through automation in creating and modifying friend groups could be enormously helpful for OSNs users.

In 2010, FeedMe [4], a content sharing web plug-in for Google Reader, has been proposed to recommend friends who might be interested in receiving a message about a topic. In this vein, in 2011, Facebook launched smart lists, human assisted lists through automation. Example lists include close friends and family groups. The interface includes a recommendation system whereby additional friends are suggested for given groups [24]. Similarly, Katango, a start-up now acquired by Google+ [18], launched a

Facebook mobile application to automatically sort friends in groups with minimal user assistance [17]. Subsequently, Amershi et al. [2] presented ReGroup, an interactive machine learning system that suggests members for the groups. In the context of private messaging systems, SocialFlows [21], an email-based application created friend groups based on the history of email communication.

Many of the mentioned automated grouping techniques employ clustering or group detection algorithms to discover friend groups in OSNs. In spite of many existing clustering algorithms, there is no gold standard for grouping members in social networks [10]. A main reason is the lack of a ‘*ground-truth*’ or gold standard template for a group. Most current evaluation metrics for clustering algorithms rely on a pre-existing ground-truth for comparison to a derived group. While some clustering algorithms such as Markov Clustering perform very well analyzing protein-protein interaction network [6], finding meaningful relationships for grouping in social networks is not straightforward. With dynamically changing relationships and networks in social media, it is not clear a single ground truth exist at any point in time.

Despite this, researchers approximate ground-truth to explore the nature of groups. Jones and O’Neill [14] applied a clustering algorithm on Facebook. They then used manually created groups as ground-truth for comparison with the automated grouping results. This approach assumes users know and can identify real groups within their structural social networks. In a similar vein, a few studies started to collect the ground-truth data from different social networks by asking people to label their groups [22, 28]. None of these studies, however, evaluate the effect of automating group detection for grouping friends by OSNs users. Rather, they collect the ground-truth data by asking people to group their friends manually and use it for evaluating clustering algorithms.

While automating friend grouping has been discussed in previous studies, to date no academic work has explored the strengths and weaknesses of automated clustering algorithms in OSNs using an interface. In this paper, we begin by applying different automation approaches on Facebook friendship networks and evaluate the groupings qualitatively and quantitatively. In the next section, we introduce the three chosen clustering algorithms used in our study.

### 3 Clustering Algorithms

In choosing a subset of clustering algorithms for our study, we explored algorithms with different input information. *Network structure* is the most common input information used by clustering algorithms. This information represents people as nodes and their friendship relations as links. Algorithms with this input information are called structure-based (or structural) algorithms. Other algorithms, called feature-based algorithms, use *nodes’ features and attributes* to detect groups. For example, these features can be age, gender, and education of people in OSNs. A third category combines these two inputs *network structure and nodes’ features*. In this paper, we focus on structure-based clustering algorithms. One advantage is the ability to interpret why the resulting groupings emerged and to compare algorithms with a consistent evaluation metric across the same network structure [10]. Furthermore, using feature-based algorithms necessitates extracting extra data from an OSN. This extraction results in very high processing time

which makes conducting studies in a limited time in the lab difficult or almost impossible.

Structured-based clustering algorithms can be further classified into three categories based on their membership attribute: (i) ‘*disjoint clustering*’ algorithms where each object can only belong to one group; (ii) ‘*overlapping clustering*’ algorithms where an object can be a member of more than one group. For example, a person may belong to different groups such as ‘Family’, ‘Main East High School’, and ‘Loves Red Sox’; and (iii) ‘*hierarchical clustering*’ algorithms which categorize objects in a multi-level structure where one group can be a subset of another group [22]. For example, cousin Joe is in a group labelled ‘Cousins’ which is a subset of a group named ‘Family’. ‘*Hierarchical clustering*’ algorithms have been used widely in social network analysis [10]. Figure 1 shows a schematic view of these clustering algorithms based on the defined membership attributes. We chose a representative algorithm from each membership category explained above for a total of three algorithms:

- *Markov Clustering (MCL)*: This algorithm is a *disjoint clustering* algorithm that uses the concept of Markov chains to simulate stochastic flows in graphs and builds a fast and scalable unsupervised clustering algorithm. MCL has a relatively high performance and is scalable [26].
- *OSLOM*: The Order Statistics Local Optimization Method (OSLOM) is an *overlapping clustering* algorithm that is among the first to account for edge weights and overlapping groups. It has a high performance and is scalable to large networks [19].
- *Louvain*: This *hierarchical clustering* algorithm uses modularity as its objective function and maximizes it using multiple heuristics to detect the groups. While this algorithm finds groups in a hierarchical manner, the lowest level of the hierarchies, which are the subgroups, are disjoint; i.e. one person cannot be a member of more than one group in a same level. The Louvain algorithm is highly accurate and has a very low computation time which makes it appealing for our study [5].

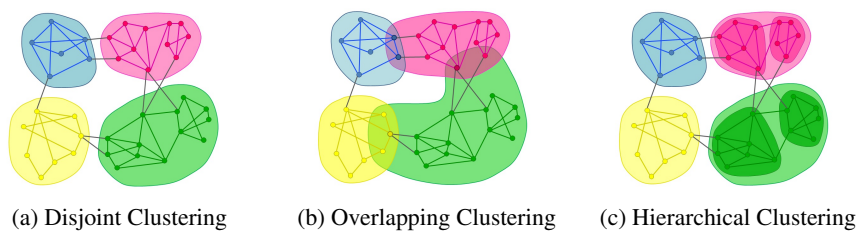


Fig. 1: Three clustering methods with different membership attributes

## 4 Method

We conducted a three part mixed methods study to better understand how social media users currently create and use groups and to evaluate how an automated approach



Fig. 2: A Snapshot of the Facebook Group Detection Application

would fit into our users' intended grouping goals. Our methodology consists of: (i) a *pre-interview* to understand existing group usage in social media; (ii) a *lab study* using a customized Facebook grouping application to understand how users perceive and modify automated groups; and (iii) a *post-interview* to explore the advantages and disadvantages of automated group creation. We recruited 18 (11 female and 7 male) participants during two months from a large Midwest university. They were from 8 different departments and ranged in age from 18-55. The participants' Facebook friendship networks ranged in size from 139 friends to 1853 friends ( $\mu = 601.7, \sigma = 367.5$ ). All the participants reported using Facebook daily (on average for the past 5.7 years) and the majority of them logged into Facebook several times a day ( $n=12$ ).

#### 4.1 The Pre-Interview

We first asked participants about basic demographics information, the social networking sites they used, and how frequently they used their favourite social networking site. We then probed them on the perceived importance of friend grouping in social networks and asked them whether they had used any friend grouping tools and why. If the participants mentioned using Facebook lists, we asked them about the type of lists they used (regular, smart or both), their goal in using Facebook lists, and the helpfulness of Facebook smart list suggestions.

#### 4.2 Facebook Grouping Application Use

For the second part of our study, we implemented a novel automatic Facebook grouping application [1]. We used Facebook API v1.0 to extract participants' friendship networks. Our application utilized the three structural clustering algorithms explained in

Section 3 to automatically group friends on Facebook. Figure 2 shows the groups created by each clustering algorithm in a separate tab. Each tab is named after the corresponding membership attribute of the clustering algorithm: disjoint, overlapping, and hierarchical. Each tab contains two panels: the *groups panel* (left side) and the *members panel* (right side). The groups panel shows the created groups by the corresponding algorithm. By clicking on a group in this panel, the members of that group are shown in the members panel. Users can move their friends from one group to another. They can also change the name of a group through both the groups and members panels. At the bottom of the group panel, there is a category named ‘*ungrouped*’ which contains any friends that the algorithm did not place into existing groups. The overlapping and hierarchical tabs offer some additional features. For instance, in the overlapping tab, moving a member from one group to another group would not result in removing the member from the first group. Similarly, in the hierarchical tab, color coding distinguishes groups at different levels of the hierarchy (see [1] for a thorough explanation of the interface).

After a brief introduction to the application, we asked participants to modify each algorithm’s automated groups considering the task of content selective sharing. As a first step, we asked them to look over each group and label it based on at least  $\frac{2}{3}$  of the group members. If a group had no meaning for them, we asked them to delete the group. When a group was deleted, its members automatically went to the ungrouped category. After the first round, participants were asked to come back and review the members of each group individually. During the review process, they were asked to move or delete members when they did not belong to a group, create new groups, or merge the existing groups as necessary. Finally, we asked them to check the members of the ungrouped category to see whether they could find a group for any of them. The participants repeated this process for each tab. To mitigate any learning effects, order effects, and bias toward a specific algorithm, we randomized the order of tabs. Due to time constraints, the participants with large network sizes ( $n > 500$ ) were asked to work on one or two of the algorithms only.

### 4.3 The Post Interview

Upon the completion of group modification in each tab and before moving to the next tab, we discussed with participants to understand how usable the interface was. We then asked them to rate the quality of the groups based on their usability before and after the modification process on a 5-point scale. We then followed up with a short semi-structured interview asking questions about each method’s performance, weaknesses and strengths. We encouraged participants to discuss any interesting or challenging points they found during the modification process in that tab. After modifying the groups in all the tabs, the participants were asked to compare the performance of the algorithms by ranking the groupings before and after the modification process (see Appendix for the detailed questions).

## 5 Evaluation

During the study, we asked participants to compare our application with the existing recommendation-based interface of Facebook lists. The majority of the participants

stated that our automated grouping interface removed the burden of verifying friends’ groups individually in comparison with the Facebook interface: “*Suggesting friends by Facebook is not user friendly as I have to add each person one by one; additionally changing a list of friends is not easy because it needs many clicks! I prefer this user interface that creates groups and then I [can] modify them. It will be faster.*” (P3). They declared that if Facebook had this interface, they might be more willing to manage their friendship network: “*if Facebook had this feature, I would probably use it. When Facebook came out, it didn’t have the list feature and then when it had it, it was hard to do it by hand. So, this version will make it easy to manage my groups of friends.*” (P9). This overwhelming preference of the proposed interface to the current recommendation-based approaches illustrates the necessity of automated friend grouping in social networks specifically when users deal with a large number of friends [8].

In the following sections, we evaluate the groups created by the algorithms to understand how well these algorithms detected users’ friendship groups. We then investigate the modified groupings of users to find out whether an automated grouping technique can bias the user’s ideal groups. Finally, we explore the group dynamics without ground-truth by using two group scoring metrics to see how well these metrics are able to identify human-curated groups.

### 5.1 Evaluating Groups and Algorithms

To assess the effectiveness of our automatic friend grouping application, we relied on both quantitative and qualitative metrics. The quantitative metrics helped us to measure the similarity of the ‘*predicted grouping*’ (i.e. the original group structure created by our application) and the ‘*desired grouping*’ (i.e. the final group structure modified by a participant), while the qualitative metrics were used to measure the level of the user’s satisfaction with the groups created by our application.

To measure the similarity of the predicted and desired grouping, we utilized a metric named BCubed, inspired by precision and recall metrics [3, 12]. For BCubed, a value of 1 represents identical groupings and 0 illustrates that none of the friends are grouped similarly in two groupings. While BCubed indicates the similarity between the predicted grouping to the desired grouping, it may not convey the user’s satisfaction level with the algorithms or our interface. For example, during the study, a few participants became confused during the modification phase and they were not able to completely create their ideal grouping. Therefore, in addition to the BCubed we asked participants to state a quality rating for each of the groups prior to modification on a 5-point Likert scale (1=poor, 5=excellent).

Algorithm	BCubed [0-1]	Participants’ Rating [1-5]
MCL (Disjoint)	0.89	3.3
Louvain (Hierarchical)	0.86	3.2
OSLOM (Overlapping)	0.78	3.1

Table 1: The Algorithms evaluation by BCubed metric and participants’ rating



Table 1 shows the participants’ Facebook friendship information and the results of each algorithm’s performance using BCubed and participant ratings. As the results demonstrate, (unlike OSLOM algorithm) both MCL and Louvain final groupings are highly similar in average to the ones participants modified. This significant similarity illustrates high accuracy of these two algorithms in detecting friendship groups in Facebook networks. In contrast to the previous work [14] in which a structural clustering algorithm (SCAN) could not find the groups of friends in OSNs with such high accuracy, our results illustrate that an appropriate structural clustering algorithm such as MCL and Louvain can detect the desired friendship groups with a significantly high accuracy while preserving human satisfaction. This outcome shows that the proper selection of a structural clustering algorithm besides including some attribute-based features of social networks (such as intimacy) can lead us to an accurate automated friend grouping approach in OSNs.

## 5.2 Automation Bias

Kelly et al. [16] investigated how different manual grouping techniques affected the final groups created by one person. They discovered that while it was possible to have an internal ‘ground-truth’ as the user’s desired grouping, the manual grouping strategies could bias the user in creating his/her desired groups. While automation has been suggested as a solution to mitigate the burdens of manual grouping, it can also introduce bias in the friend grouping process. To examine whether such bias exists in our automated grouping techniques, for each participant, we compared the desired groups that emerged from the MCL predicted groups with the desired groups that resulted from the Louvain predicted groups. The comparison was performed using the BCubed metric and revealed that the MCL and Louvain desired groups created by the same participant are different from each other by 14% on average. This difference suggests that automated techniques (i.e. MCL and Louvain) used for generating the predicted groups can influence the desired groups created by a participant. We did not compare the desired groups created from OSLOM with the desired groups produced by the MCL and Louvain modification since OSLOM predicted groups are not disjoint.

In order to understand the possible causes of the bias introduced by the automation techniques, for each participant we carefully examined the difference between the MCL desired groups and the Louvain desired groups. To this end, for each group from a set of desired groups, we found its corresponding group in the other set of desired groups. Then, we looked over the groups with the most difference in two sets of desired groupings. Investigating these groups, we found that this difference is caused by four main factors:

1. *Following What Algorithms Create:* Some participants stated that if an algorithm did not find a specific group, they would not create that extra group. For example, one of the participants mentioned that one algorithm put his ‘church’ friends in a separate group. If he had manually created groups, he suspected he would not have considered a ‘church’ group. He then admitted the group made sense, he liked it, and kept it. Such examples demonstrate that automating the friend grouping process influences users to follow what algorithms seed.

2. *Existing Hierarchical Structures in Social Relationships*: One potential cause of inconsistency between the two sets of desired groups created by the same participant, is the difference in the hierarchy levels of the initial predicted groups. For example, while MCL might detect a group that a participant would call “university”, Louvain might divide this group into smaller groups that the same participant would label using criteria such as entrance year or closeness. Therefore, after the participant was done with the modification process, the desired groups from these two algorithms would differ (see table 2) .

Automation Bias Reasons	P#	MCL Desired Groups	Louvain Desired Groups
Hierarchical Structures	P12	US High School	US HS 2010, US HS 2011
	P14	ECO	ECO, ECO close, Others
	P15	Facebook	Facebook, Facebook interns, University CS
Friends with Multiple Roles	P5	Family	Family, Brother’s Friends
	P11	Industrial design	Industrial design, Roommates, Art and design
	P15	Chicago friends	Chicago friends, University Other
User’s Uncertainty	P7	Not close (University)	Not Close Uni Friends, Average Uni Friends
	P9	April’s Family	April’s Family, Family and Family Friend
	P15	Family	Family friends, Un-Grouped

Table 2: *Examples of Automation Bias Causes.*

3. *Having Friends With Multiple Roles*: Some of our participants had a number of friends with multiple roles, but they could assign these friends to only one group due to MCL and Louvain’s disjoint membership constraint. Our participants’ decisions on the most appropriate group for this type of friend were affected by the available predicted groups. For example, when a friend was a member of family and also a classmate in the university, the participant assigned this friend to the predicted group which could be the ‘family’ group in one algorithm and the ‘university’ group in the other algorithm. More cases are shown at table 2.

4. *User’s Uncertainty*: One of the main issues in the friend grouping process was the participants’ uncertainty when identifying or creating groups for some friends. For example, one of the participants started to make a ‘Bay Area’ group and decided to make it more specific based on different organizations (Facebook, Yahoo and ...). She eventually became confused with the organization of these groups and gave up. This confusion came from the uncertainty in identifying the right group. In another case, we found some participant were unable to distinguish the intimacy levels between some friends. For example, while a participant created a group named ‘closer friends (University)’ after modifying the Louvain predicted groups, she divided this group to two groups of ‘Average University Friends’ and ‘Close University Friends’ in the modification process of the MCL predicted groups. Table 2 shows more examples of uncertainty.

### 5.3 Exploring Group Dynamics without Ground Truth

In this study, we used the BCubed metric to compare the predicted groups generated by an algorithm to the desired groups made by a participant. In most real-world cases,

we cannot access the desired groups or so called ground-truth. Therefore, various group scoring metrics have been defined to evaluate groupings in the absence of ground truth. These metrics are grounded in the general definition of a group—a group has many connections between its members and few connections to the rest of the network. Recent work evaluated these metrics by applying them to social, collaboration, and information networks where the nodes had explicit group memberships. Of the thirteen evaluated metrics, we chose the two with the best consistent reported performance in identifying ground-truth communities: Conductance and Triad-Participation Ratio [20, 28].

- *Conductance*: This metric measures the fraction of total links of a cluster that point outside the cluster. Since a group by definition has more connections between members than outside, a conductance of 0 represents an ‘ideal’ group with no connections to the rest of the network; a conductance of 1 implies no connections within that grouping [20].
- *Triad-Participation Ratio (TPR)*: TPR metric is the fraction of members in a group that belong to a triad, a set of three connected nodes, inside the group. Unlike conductance, a higher TPR represents a tighter group [28].

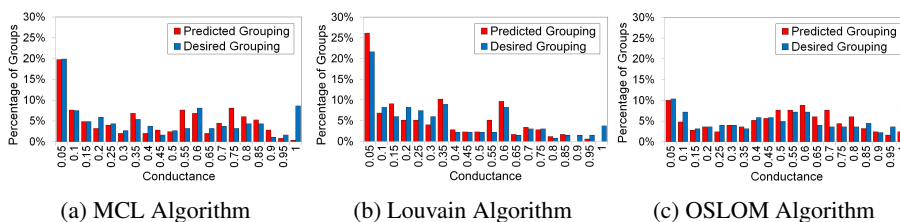


Fig. 3: Histogram Percentage of Groups at various Conductance Values.

We measured these metrics over predicted and desired groups to compare their values before and after users’ modification. We hypothesized that the desired groups will have a lower conductance and higher TPR with respect to the predicted groups. To test our hypothesis, we calculated these metrics for predicted and desired groups produced by the three clustering algorithms. We found that the TPR metric increased significantly after the modification process as it was expected. However, the number of groups with high conductance ( $[0.80-1]$ ) increased by 10% (Figure 3). That is, the number of groups with almost no inside connections between members increased after the modification process. To further explore this unexpected result, we investigated the groups which their conductance value increased after the modification process. We also coded the transcripts from our interviews where participants described their grouping process. We found out that some of our participants put some of their friends that were not linked together in one group. We found four categories of phenomena that explained this increase of conductance:

*Others*: This category contained the friends that participants did not care to or could not easily group. One of the participants drew Figure 4 to illustrate her grouping model.

As she explained, she saw her online friendship network as a network with three layers: (i) close friends; (ii) regular friends; and (iii) Others. She stated that he did not want to spend time to create groups for the ‘others’ layer.

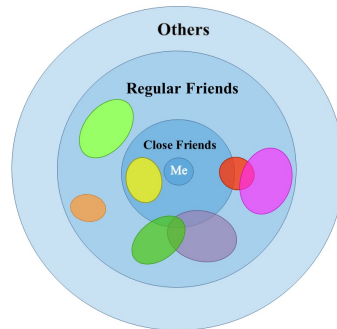


Fig. 4: A Participant’s Rendering of Friend Categories

Another participant described the people in her ‘others’ group: “*In social media, I don’t know these people very well as I meet them online and I have no more relations with them...*”. Some examples of the groups which reside in this category are shown in Table 3. The common attribute between the individuals in these groups is ‘not being important to be in a labelled group’. Therefore, there is a lower chance for the members of these groups to be connected.

*Functional Ties:* Facebook is a social networking site, yet some people use it to maintain connections that are not reciprocally social [14]. These connections were added for professional or functional reasons. For example, one of our participants made a group labelled ‘political’ and said this group contained important people in policy whom he follows. However, the members of this group were not mutual friends in Facebook since they were from different political backgrounds. This resulted in a higher conductance in this group. Other examples of functional ties are shown in Table 3.

*Indirect Friends:* Our participants treated some of their online friends as indirect friends and consequently grouped them as friends of other friends. One of our participants made a group labelled ‘friends of friends’: “*I made a group named ‘friends of friends’ that contains people who friended me but are my friends’ friends but they might not know each other even [if] they are in one group!*” There are similar examples in Table 3 such as ‘Friend’s siblings’ where the members in the group may not be connected. These examples explain the high conductance in these groups containing indirect friends.

*Temporal Ties:* These are friendships that are bounded in time. Many of our study participants created groups such as ‘*People I worked with/talked to once and never again*’. One participant labelled a group ‘*We win competitions and hackathons for silly ideas*’ and described it as a group of people he knew during a contest. Other examples of temporal ties can be seen in Table 3. The short-term temporal tie relationships increase the probability for fewer connections in an online space such as Facebook.

Category	Group Name
<b>The Others Group</b>	University friends who don't fit other groups (1), University other (1), Don't know (1), People I don't remember ever having talked to (0.91), Others [0.83 - 0.97], Un-Grouped [0.84 - 1]
<b>Functional Ties</b>	Advertising/Journalism people I met from totally different places (0.84), Political (0.87), Old Teachers (1), Bloggers and Organizations (0.83-0.94)
<b>Indirect Friends</b>	Friend's siblings (0.87), Stevenson close friends (0.93), Brian's friends (0.87), Sisters friends (1), Friends of Friends (1), Met via Sibs (0.95)
<b>Temporal Ties</b>	People I worked with/talked to once and never again (0.83), We win competitions and hackathons for silly ideas (0.84), Vineyard (0.95), Habitat for Humanity No Builds (0.89), Summer University (1), Old church (0.83)

Table 3: The desired groups with high conductance - () shows the conductance of each group

The different characteristics of 'The Others', 'Functional Ties', 'Indirect Friends', and 'Temporal Ties' are challenging for group scoring metrics such as conductance. The conductance and other similar metrics assume intense inside group connections, however, some of the groups our participants labelled do not fit the traditional definition of a group. This suggests that for OSNs, we should explore alternate group scoring metrics compatible with the dynamic groups that exist in these networks.

## 6 Discussion

From the three clustering algorithms that we used in our study, MCL and Louvain performed well in terms of accuracy and human satisfaction. This result suggests that structure-based clustering algorithms such as MCL and Louvain are effective in detecting groups in OSNs. However, these algorithms do not consider some important features such as intimacy or interaction between friends. During our study, many participants said that the groups generated by the algorithms would have been more useful if they were able to separate their close friends from other friends or split some of the groups to smaller groups based on intimacy. However, since the applied algorithms in the study were structure-based, they did not have the required information to detect these types of groups. This finding which corroborates previous studies [11, 14] demonstrates the necessity of adding important factors such as intimacy and interaction between friends to the current structure-based clustering algorithms.

Although the participants preferred our automated friend grouping tool to the current recommendation-based interface of Facebook smart lists, this automation introduces bias in the friend grouping process. While this bias could also exist in recommendation-based tools, creating fully populated groups from onset with our automated approach could increase it. However, we believe this bias can be reduced. For example, having both hierarchical and overlapping membership attributes for supporting sub-groups and friends with multiple roles simultaneously can mitigate this bias.

In our study, participants did not care to group some of their contacts; we labeled these contacts 'others'. We believe an effective clustering algorithm should be able

to find and prune this group of contacts before starting to group the friends. Pruning contacts helps increase the accuracy of the clustering algorithm when detecting the actual friendship groups. Furthermore, our results revealed that there were some other types of groups besides 'the others group' (functional ties, indirect friends and temporal ties) which current group scoring metrics cannot identify. It would be fruitful to probe alternative metrics which are compatible with such human-curated groups in OSNs.

A limitation of our study is the small sample of university students. We look forward to collecting data from additional OSN users with more diverse friendship networks. Another challenge was the time it took participants with large numbers of friends to use the three different interfaces. On average, participants completed the study in 114 minutes. This length of time could result in human fatigue and consequently, human error during the modification process. To lessen this effect, we adjusted the number of automated approaches based on the participant's number of friends. Sampling friends in a uniform way to reduce the time while still providing significant results could be a fruitful approach for future work. Finally we asked participants' perceptions of groupings rather than having them use the created groupings in a real world task. Our subjects were told to imagine groupings for selectively sharing a message/image in Facebook. Future work should observe users sending specific content using the grouping approach described in the paper.

## 7 Conclusion

Given the significant cost of manual grouping in OSNs, this work takes a step toward providing an automated friend grouping tool that applies three different clustering algorithms on Facebook friendship networks. Studying this tool, we found that users preferred our automated friend grouping tool to the current recommendation-based Facebook smart lists. We compared the three clustering algorithms using quantitative and qualitative evaluation methods. The evaluation results showed that the MCL and Louvain algorithms performed well in terms of accuracy and human satisfaction. While our automated friend grouping tool was well received by the participants, comparing the desired groups created by two different algorithms illustrated a significant bias in the automation approach. We believe future work should address educating users of these biases in their algorithmic interfaces. In our analysis of group composition before and after the modification process using two group scoring metrics, we found four categories of groups which do not satisfy the traditional definition of a networked group. This suggests that more exploration is needed and perhaps new metrics are necessary for understanding groupings of real world social connections. Grounded in our findings, we presented suggestions for designing future automated friend grouping tools. This work is a promising step toward designing an automated friend grouping framework for OSNs' users which can help manage their contacts efficiently.

## 8 Acknowledgements

This research is funded by the NSF (#0643502) and the Center for People and Infrastructures at the University of Illinois at Urbana-Champaign.

## References

1. Mygroups. <http://social.cs.uiuc.edu/projects/MyGroups/CDA/index.php/frontend/intro>.
2. S. Amershi, J. Fogarty, and D. S. Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of CHI 2012*, 2012.
3. E. Amigó, J. Gonzalo, J. Artilles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, Aug. 2009.
4. M. S. Bernstein, A. Marcus, D. R. Karger, and R. C. Miller. Enhancing directed content sharing on the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 971–980, 2010.
5. V. Blondel, J. Guillaume, R. Lambiotte, and E. Mech. Fast unfolding of communities in large networks. *J. Stat. Mech*, page P10008, 2008.
6. S. Broh e and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488, 2006.
7. A. Carr. Facebook’s New Groups, Dashboards, and Downloads Explained, Oct 2010. <http://www.fastcompany.com/1693443/facebook-s-new-groups-dashboards-and-downloads-explained-video>.
8. M. Eslami, A. Aleyasen, R. Zilouchian Moghaddam, and K. Karahalios. Evaluation of automated friend grouping in online social networks. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, pages 2119–2124. ACM, 2014.
9. W. K. Estes. *Classification and Cognition*. Oxford University Press, 1994.
10. S. Fortunato. Community detection in graphs. *CoRR*, abs/0906.0612, 2009.
11. E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220. ACM, 2009.
12. J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers Inc., 2000.
13. M. L. Johnson. Toward usable access control for end-users: A case study of facebook privacy settings. *PhD Dissertation University of Columbia US*, 2012.
14. S. Jones and E. O’Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 9:1–9:13. ACM, 2010.
15. S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi. Talking in circles: selective sharing in google+. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1065–1074, New York, NY, USA, 2012. ACM.
16. P. G. Kelley, R. Brewer, Y. Mayer, L. Cranor, and N. Sadeh. An investigation into facebook friend grouping. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, INTERACT'11, pages 216–233, 2011.
17. J. Kincaid. Kleiner-Backed Katango Organizes Your Facebook Friends Into Groups For You, Jul 2011. <http://tcn.ch/10qQ7A6>.
18. J. Kincaid. Google Acquires Katango, The Automatic Friend Sorter, Nov 2011. <http://tcn.ch/1gtN9jD>.
19. A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 2011.
20. J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proc. of the 19th international conference on World wide web*, WWW '10, pages 631–640. ACM, 2010.
21. 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 16th international conference on Intelligent user interfaces*, IUI '11, pages 83–92. ACM, 2011.

22. J. McAuley and J. Leskovec. Discovering social circles in ego networks. *CoRR*, abs/1210.8182, 2012.
23. B. Ross. Improved Friend Lists, Sep 2011. <http://on.fb.me/1rbm98o>.
24. M. Slee. Friend lists, Dec 2007. <http://on.fb.me/1oHzyp2>.
25. B. Stone. There's a list for that, Oct 2009. <https://blog.twitter.com/2009/theres-list>.
26. S. M. Van Dongen. Graph clustering by flow simulation. *PhD Dissertation University of Utrecht, The Netherlands*, 2000.
27. J. Watson, A. Besmer, and H. R. Lipford. +your circles: sharing behavior on google+. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 12:1–12:9, New York, NY, USA, 2012. ACM.
28. J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3:1–3:8. ACM, 2012.

## 9 Appendix: Post-Interview Questions

1. At first glance, how would you rate the quality of clusters created by this method? (1= poor, 5= excellent)
2. How well are you satisfied with the final groupings you made after the modifications? (1= Not very, 5 =Very)
3. How would you rate the groups created here by their usability? e.g. this grouping is useful for text messaging, announcing special events or ... (1= unusable, 5= usable)
4. How comfortable were you with the interface of this method? e.g. working with groups, moving friends, ... (1= Not very, 5= Very)
5. What worked well about this method? Can you give specific scenarios?
6. In what circumstances did this method not work well? Can you give specific scenarios?
7. If you decide to continue working on this grouping, is there any group you want to work on to make it better?
8. How cautious and accurate do you think you made your groups?