Non-Sharing Communities? An Empirical Study of Community Detection for Access Control Decisions

Gaurav Misra
Security Lancaster
Lancaster University, UK
Email: g.misra@lancaster.ac.uk

Jose M. Such
Security Lancaster
Lancaster University, UK
Email: j.such@lancaster.ac.uk

Hamed Balogun
Security Lancaster
Lancaster University, UK
Email: h.balogun@lancaster.ac.uk

***Abstract*—Social media users often find it difficult to make appropriate access control decisions which govern how they share their information with a potentially large audience on these platforms. Community detection algorithms have been previously put forth as a solution which can help users by automatically partitioning their friend network. These partitions can then be used by the user as a basis for making access control decisions. Previous works which leverage communities for enhancing access control mechanisms assume that members of the same community will have the same access to a user's content, but whether or to what extent this assumption is correct is a lingering question. In this paper, we empirically evaluate a goodness of fit between the communities created by implementing 8 community detection algorithms on the friend networks of users and the access control decisions made by them during a user study. We also analyze whether personal characteristics of the users or the nature of the content play a role in the performance of the algorithms. The results indicate that community detection algorithms may be useful for creating default access control policies for users who exhibit a relatively more *static* access control behaviour. For users showing great variation in their access control decisions across the board (both in terms of number and actual members), we found that community detection algorithms performed poorly.**

1. Introduction

Social media has become synonymous with communication in daily life for most of us. On Facebook alone, over 1 billion users[1] share over 300 petabytes of personal information.[2] Social media users interact with people representing various facets of their life such as work, family, education, etc. In such a scenario, it is essential for them to be able to control access to their information. It has been repeatedly found that social media sites fall short of providing usable access control mechanisms to their users and users have to struggle as a result [1].

1. http://investor.fb.com/releasedetail.cfm?ReleaseID=924562
2. https://research.facebook.com/blog/1522692927972019/facebook-s-top-open-data-problems/

It has also been found that users visualize their friend network in the form of partitions or sub-structures [2]. This has led to the development of "Group-based Access Control" models which allow the user to leverage communities of friends within their network to make access control decisions. Mainstream social media sites such as Facebook and Google+ have made an effort to implement this model and assist users by creating Lists [3] and Circles [2] respectively. Most of such communities are based on the "life-facets" the particular friends represent. Family, co-workers, acquaintances are some examples of such communities [2]. However, it has been observed that users do not usually employ these mechanisms when making access control decisions [3]. These communities are formed based on the input of the users and the relationship as defined by the pair of users on the social media site. A family list, for example, contains all the contacts of a user who are explicitly mentioned as being family members on the site. Thus, the responsibility of maintaining the appropriateness of these communities lies solely on the users which puts a *cognitive burden* on them. Moreover, many users do not provide accurate information about their location, workplace or relationship with another individual (e.g: family members may not be explicitly listed as such on the social media site) which would further undermine the accuracy of such mechanisms.

Another method of partitioning a user's friend network is by employing network based community detection algorithms. Many proposed privacy protection mechanisms suggest the use of these algorithms to create communities in a user's friend network and use them as a basis for making access control decisions [4], [5], [6], [7]. Such access control mechanisms, however, have an underlying assumption that members of the same community will have the same access control decisions (allow/deny), assumption that has not been adequately examined yet. Moreover, most of the previous works in this area implement only one community detection algorithm and present the results based on that algorithm, so a comparison of different algorithms is also missing. In this paper, we bridge this gap by providing an empirical study of eight community detection algorithms

3. https://en-gb.facebook.com/help/135312293276793/

with respect to the access control decisions made by users. We examine a goodness of fit between the communities created by the algorithms and the access control decisions made by users in an actual sharing scenario during a user study. The major contributions of this work are:

- Provide an empirical study of 8 community detection algorithms examining a goodness of fit between communities created by the algorithms and the access control decisions made by users
- Examine the effects of each individual user as well as the categories of the disclosed items on the performance of the algorithms
- Identify the type of access control behavior for which communities created by the algorithms provide the best fit

Our analysis found that even the best performing algorithm of those evaluated, Clique-Percolation Method (CPM), may not produce communities which are accurate enough to be used directly for access control decisions by all of the users. We found, however, a scenario where a good fit seems to exist between the communities produced by the algorithm and the access control decisions made by a type of users. This was observed for about two-thirds of our participants (20 out of 31) who exhibited a relatively more *static* access control behaviour by denying access repeatedly to the same friends — though not always granting access to all the others. Other personal characteristics of the individual users (like Gender, Age, etc.) as well as the nature of the photo (as reported by the users themselves) were not found to have a major influence on the results.

2. Related Work

There have been a number of works that proposed using community detection algorithms to facilitate the definition of access control policies [4], [5], [6], [7]. The underlying assumption is that members of the same community will have a similar access control decision (allow/deny). One particular way of defining an access control policy is to ask the user to define access control decisions for one or some of the members of each community, and implement these decisions on the other members not explicitly mentioned [4], [7]. Such enhancements can be used to reduce users' effort in defining access control policies, as they do not have to take decisions for each and every individual friend in their entire friend network. It has also been found that users prefer to select their audience from predefined communities as compared to their entire friend lists [5] and that community membership could also help in learning the privacy policies of individual users [6]. The problem with much of the existing work in this area is that the underlying assumption that members of the same community will be treated similarly has not been adequately examined empirically. This assumption depends heavily on the goodness of fit between a user's conception of their audience and the communities created by the algorithm. Moreover, most of these works implement only one algorithm for their experiments and a comparison and an evaluation of community detection algorithms in an access control context is absent. Such a comparison is essential before making any conclusions about the quality of fit of communities created by the algorithms with access control decisions made by users. Fogués, et al. [8] did test three community detection algorithms, but algorithms were compared in terms of whether the communities created would be accepted as such by users, not in terms of their direct goodness of fit with access control decisions. In this paper, we bridge this gap by empirically evaluating 8 community detection algorithms with user data and activity about practical access control decisions.

An alternative method of creating communities is to use profile vectors of social media profile data to create "ego networks". ReGroup [9] uses "identifying features" for all the Facebook friends of a user to provide suggestions of audience members dynamically using a machine-learning approach. This "on-demand" community creation is based on calculating similarity of Facebook profiles for all of the user's friends. McAuley, et al. [10] present an algorithm which uses a combination of both network and node information to create circles from an 'ego' network of a particular user (where the user is the 'ego' and his friends are the 'alters'). A user profile vector is created to capture the similarity in two alters. Squicciarini, et al. [11] use profile vector similarity to identify sub-structures within a user's friend network which are then used to mine appropriate privacy policies for a user under the assumption that similar users have similar privacy policies. However, all profile vector based approaches have been found to have a major limitation due to the problem of missing information, as all users do not provide the same amount of information in their social media profiles [9]. In some cases, even if the information is available, it is incorrect as some users choose to provide incorrect information on their social media profiles. Moreover, many social media sites do not allow to get the profile attributes of users for privacy reasons, e.g. from April 2015, Facebook has removed support for FQL and now only graph API can be used, so that an application can only access the attributes of users that utilize it[4], which challenges the reproducibility and validity of evaluating approaches that depend on profile attributes.

3. Method

In a nutshell, we implemented 8 community detection algorithms and applied them to the friend networks of the participants. The communities created by these algorithms were stored in a secure database. During the experiment, the participants were required to select audiences which were then compared with the communities created by the various algorithms in order to evaluate their goodness of fit in an audience selection scenario.

4. https://developers.facebook.com/docs/apps/changelog

3.1. Algorithms Considered

Most of the commonly used community detection algorithms use network information and detect *cliques* or *clusters* based on some network properties. Different methodologies followed by network based community detection algorithms, as outlined by Papadopoulos, et al. [12] are: *Cohesive Subgraph Discovery, Vertex Clustering, Community Quality Optimization, Divisive Algorithms* and *Model based Algorithms*. We chose the following 8 algorithms to represent these five categories and different complexity (see Table 1) and implemented them using iGraph [13] and SNAP libraries [14]:

- ***Fastgreedy*** [15]: This is a bottom-up hierarchical algorithm which tries to optimize modularity in a "greedy" manner. It is based on *Community Quality Optimization* which relies on hierarchical agglomeration for detecting communities.
- ***Walktrap*** [16]:This is a *Vertex Clustering* based algorithm which is based on identifying similarities between vertices based on random walks.
- ***Infomap*** [17]: It is a *Model based* algorithm which tries to find the cluster structure that results in the lowest possible cluster encoding cost.
- ***Girvan-Newman*** [18]: This *Divisive* algorithm progressively removes edges of a network based on an edge-betweenness measure until disconnected clusters emerge.
- ***Label Propagation*** [19]: This *Model-based* method assigns labels to every node and proceeds iteratively and re-assigns labels to nodes in a way that each node takes the most frequent label of its neighbors in a synchronous manner. The method stops when the label of each node is one of the most frequent labels in its neighborhood.
- ***Leading Eigenvector*** [20]: This algorithm is also based on *Community Quality Optimization* and it optimizes modularity of the network by splitting the graph progressively. The split is determined by the eigenvector of the modularity matrix.
- ***Multi-Level Community*** [21]: It is based on *Community Quality Optimization* by measuring and optmizing modularity using iterative heuristic schemes.
- ***Clique-Percolation Method*** [22]: It is based on *Cohesive Subgraph Discovery* approach where the algorithm relies on discovery of pre-defined sub-structures. The sub-structures for CPM are called *k-cliques* which correspond to complete (fully connected) sub-graphs of *k* nodes. *k* is defined prior to running the algorithm and we used the default value of $k = 2$ in order to eliminate any overlap between communities.

3.2. Participants

This research experiment was conducted at Lancaster University after being approved by the Research Ethics Committee of the university. The participants were recruited primarily from among the staff and students of the university. Details about the privacy implications and the overall objectives of the experiment were communicated to the registered participants and it was conducted only after the explicit consent of the participants. All participants were compensated with £10 for their involvement in the study.

TABLE 1: Algorithms considered - complexities from [12], [23]

Algorithm	**Type**	**Complexity**
FastGreedy (FG)	Community Quality Optim.	$O(n \log^2 n)$
Walktrap (WT)	Vertex Clustering	$O(n^2 \log n)$
Infomap (IM)	Model Based	$O(n \log n)$
Girvan-Newman (GVN)	Divisive	$O(n^3)$
Label Propagation (LP)	Model Based	$O(n)$
Leading Eigenvector (LEV)	Community Quality Optim.	$O(n^2 \log n)$
Multi-level Community (MC)	Community Quality Optim.	$O(n)$
Clique-Percolation (CPM)	Cohesive Subgraph Discovery	$O(exp(n))$

Participants were included if they had a Facebook profile and had uploaded at least 10 photos on Facebook prior to the experiment. We had 31 participants, 17 males and 14 females. The age distribution was: 7 people were 20-24 years old, 13 people were 25-29 years, 10 were 30-34 years and 1 participant was above 35 years. The total number of friends (combining lists of all users) was 11726. The average size of friend network in our sample was 378 (S.D = 233, median = 349). The largest number of friends a user had was 991 while the smallest was 59.

3.3. Experiment

Participants used a Facebook application which was specifically designed and developed by the authors for this experiment. The application was built using the Facebook API to fetch information from the participants' profiles and their friend connections. All this data was then stored in secure databases for subsequent analysis.

Five photos were randomly downloaded from the participants' Facebook profiles by the application to be presented to the user for audience selection. In addition, the participants were asked to select and bring 5 other photos which they hadn't yet uploaded on Facebook. This was done to avoid a scenario where a user selects an audience for a photo during the study for which they had already received comments and likes before the user study as that may have influenced their choice of audience members. The participants were also advised to choose photos which were personal (either included them or a family member) or considered sensitive so that they had a privacy implication. The different stages of the user study were:

1) The participants logged into the application using their Facebook credentials. They were then alerted about the data that would be accessed by the application and asked for explicit permissions before moving on.
2) The participants were shown 10 photos (5 from Facebook and 5 they brought as detailed earlier) sequentially on the screen, each on an individual page. They were asked to select categories for the photos from a predefined list of 15 popular photo categories on Flickr, tag friends and select the audience for each photo. The friend list was shown alphabetically to the

participants to imitate the organization Facebook uses to show friend lists to its users. The participants were instructed to select each and every friend that they would want to grant access to the photo and were explicitly told that any friend who was not selected would be denied access to the photo.

3) Once participants selected all the categories, tags and audiences, their selections were stored in the database.

As mentioned earlier, each participant had to make an access control decision for a particular friend for each photo. Thus considering each of our 31 participants had 10 photos and varying number of friends, we had *79010* access control decisions in total in our dataset.

After the completion of the data collection during the user study, the community detection algorithms were implemented on the friend networks of all users to create communities. These communities were then compared with the selections made by the users to examine a goodness of fit using various evaluation criteria defined below.

3.4. Evaluation Criteria

We used three criteria to examine a goodness of fit between communities produced by the algorithms and the access control decisions taken by users. The criteria are defined such that they acknowledge the willingness of users to share *selectively* in communities [2] but also account for the *effort* required from the user to modify the communities into an audience for their content [8].

For the definition of each criterion, we denote one particular user with a set of friends $U = \{u_1, \ldots, u_N\}$ that an algorithm of the ones evaluated partitions into a set of communities $\mathbb{C} = \{C_1, \ldots, C_M\}$, such that $\bigcup C_i = U$ and $\bigcap C_i = \emptyset$; together with a user's photo p and the audience the user selected for that photo $A_p \subseteq U$.

Number of Communities per Audience. Intuitively, an algorithm is considered more useful if the user needs to select from a small number of communities to build the entire audience for a photo. Thus, the algorithm with a low average number of communities to complete an audience is considered a better fit for audience selection. For example, if a user selects 60 members in an audience for a particular photo and 15 are from community A, 25 from community B and the other 20 from community C according to the communities created by Fastgreedy algorithm, the value for this metric will be 3 for Fastgreedy for that particular photo.

Definition 1. Given the set of friends U, the set of communities $\mathbb{C}$, the particular photo p, and the audience for the photo A_p, the number of communities per audience is:

$$G_p = |\{C \mid C \in \mathbb{C} \wedge \exists u \in A_p, u \in C\}|$$

Ratio of audience in largest community. If the communities produced by an algorithm can be readily used to create an audience, then the burden on the user is minimized. Thus, we calculate the percentage of audience members belonging to the largest community represented in the audience.

Definition 2. Given the set of friends U, the set of communities $\mathbb{C}$, the particular photo p, and the audience for the photo A_p, the ratio of audience in largest community is:

$$R_p = \max_{C \in \mathbb{C}} |\{u \mid u \in A_p \wedge u \in C\}|$$

Penalty for Exclusion. When the user is employing the communities produced by the algorithms for audience selection, it is possible (and probable) that he would need to exclude some of the friends from some of the communities to create an audience of his choice. After all, community membership does not guarantee that all friends in the same community would always be treated in a similar way by the user. Such an exclusion from an audience will require effort from the user. Thus, the algorithms need to be evaluated with a metric that measures the number of friends in a given community who were not included in the audience for a particular photo.

Definition 3. Given the set of friends U, the set of communities $\mathbb{C}$, the particular photo p, and the audience for the photo A_p, the penalty for exclusion is:

$$E_p = \sum_{C \in \mathbb{C}} |\{u \mid u \notin A_p \wedge u \in C \wedge \exists u_2 \in A_p, u \neq u_2 \wedge u_2 \in C\}|$$

4. Results

4.1. Overall Results

We first discuss the overall results of the evaluation of the community detection algorithms using the evaluation criteria defined earlier. For each user, the value of the metric is averaged across 10 photos and then aggregated for all users for each algorithm.

Communities per Audience. From the results in Fig 1, it is clear that Clique Percolation Method (CPM) performs slightly better than the other algorithms. The results indicate that a user needs to traverse through less than 4 communities (3.83) on average to complete their audience selection if using the communities produced by CPM. Multilevel (4.84), Leading Eigenvector (4.96), Label Propagation (5.12) and Fastgreedy (5.40) produce communities such that a user may need to access approximately 5 communities to complete the audience. Infomap produces the least impressive performance with respect to this metric (8.22).

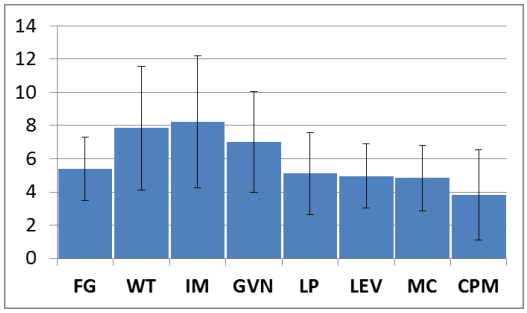

Figure 1: No. of communities required to create audience

We wanted to evaluate whether there was any signifcant difference in the performance of the algorithms with respect to each of the evaluation metrics. First, we conducted a Kolmogorov-Smirnov test (KS Test) [24]. We found that our dataset was significantly deviant from normal distribution ($p<.001$). Therefore, we were unable to conduct an ANOVA which requires samples to be normally distributed. Thus, we required a Kruskal-Wallis H Test [25] which checks for significant difference between treatments (the 8 community detection algorithms in our case) and does not require the dataset to be normally distributed. For communities per audience, the Kruskal-Wallis test suggests that there is a significant difference between the algorithms ($p<.001$). To identify the source of difference, we conducted a Mann-Whitney U Test [26]. We found the 8 algorithms could be divided into three sets:

- *CPM*, which performs significantly better ($p<.001$) than all other algorithms
- *Fastgreedy, Label Propagation, Leading Eigenvector* and *Multi-level Community* had no significant difference between their performance but were significantly better ($p<.05$) than the algorithms mentioned below
- *Walktrap, Infomap* and *Girvan-Newman* performed significantly worse than the above mentioned algorithms but did not have any significant difference between them

Ratio in Largest Community. Looking at Fig 2, we find that CPM is the best performer compared to all other algorithms. On an average, 74% of the audience can be selected from a single CPM community. All other algorithms produce similar performance where approximately half the audience can be selected from the same community.

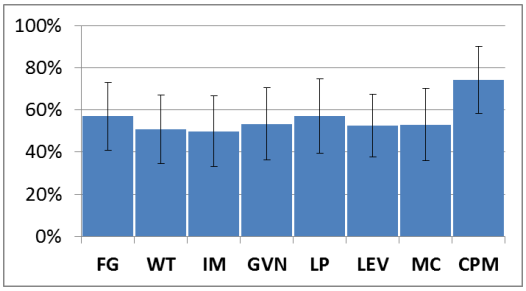

Figure 2: Ratio of largest community in audience

The Kruskal-Wallis test for this metric also showed a significant difference between the different algorithms. The Mann-Whitney U Test revealed that CPM had a significant difference in performance with all other algorithms. None of the other 7 algorithms had a statistically significant difference in performance when compared with each other.

Penalty for exclusion. The penalty values in Fig 3 signify the number of users the participant would have had to exclude to obtain the desired audience from the communities created by the algorithm. As can be seen from the results, CPM performs marginally better than all other algorithm with a penalty value of nearly 171.

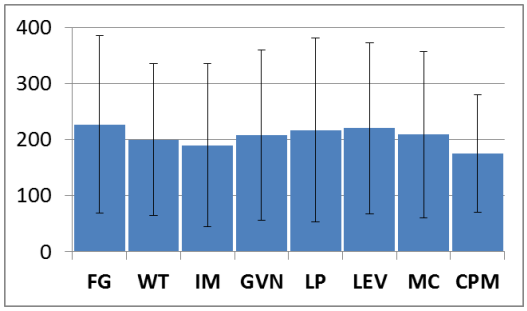

Figure 3: Penalty for exclusion

Kruskal-Wallis test for penalty for exclusion revealed that there was no significant difference between the 8 algorithms for this metric ($p = 0.872$). This can also be anticipated by looking at the descriptive statistics shown in Figure 3 where the performance is only marginally different.

Looking at the overall results of the evaluation of the algorithms according to our defined evaluation metrics, we observe a large standard deviation, particularly for penalty for exclusion, for almost all the algorithms. This necessitates a deeper analysis into other factors which might have influenced the results.

4.2. Effect of Individual

Personal Characteristics. Table 2 shows Pearson correlation for the performance of the algorithms according to the three evaluation metrics with respect to gender and age of the participants as well as their size of friends network and average audience size per photo.

It can be seen that size of the friend network has negligible effect on any of the metrics. We see that gender of participant has moderate to high negative correlation with ratio in largest community for all algorithms except CPM. We coded males as 0 and females 1 for the binary correlation analysis and hence it can be concluded that males are more likely to have a higher ratio of their audience in the largest community as compared to females. Looking at the age of participants, it has a moderately positive correlation on communities per audience and moderately negative correlation with ratio in largest community for all algorithms except LP and CPM. Average audience size positively effects the number of communities in audience (more people, more communities required) for all algorithms. Its correlation is highest for Infomap. We find that the penalty for exclusion is not affected by any of these factors substantially. In terms of particular algorithms, we find that CPM produces minimal correlation for most factors and hence can be considered the most resilient to such individual characteristics.

Static Access Control Decisions. We also sought to examine whether changing access control decisions with respect to an individual friend has any effect on the usefulness of communities. We looked at the friends for each user where the access control policy remained constant for them. That is, they were either selected in the audience for all 10 photos by the user or they were excluded from the audience in all photos.

TABLE 2: Correlation of performance of algorithms with respect to characteristics of individual

	FG	WT	IM	GVN	LP	LEV	MC	CPM
Comms.								
Gender								
Age	+	+	+	+		+	+	
Size								
Avg Aud	+ +	+ +	+ + +	+ +	+ +	+	+ +	+
Ratio								
Gender	− −	− −	− −	− −	− −	−	−	
Age	−	−	−	−		−	−	
Size								
Avg Aud	−	−	−	−	−	−	−	
Penalty								
Gender								
Age								+
Size								
Avg Aud								

Strong Correlation (+ + + or − − −) : Coefficient> 0.7
Moderate Correlation (+ + or − −) : Coefficient between 0.5 and 0.7
Weak Correlation (+ or −) : Coefficient between 0.3 and 0.5
Negligible Correlation (no symbol) : Coefficient < 0.3

We found that only 3 out of the 31 users had any friends who were *always selected* in the audience for all of the 10 photos. One user had 2 such friends while the other two users had 1 friend each in this category. The other 28 users had no friends who were constantly selected in every photo. This negligible proportion of permanently selected friends rules out the possibility of creating a subset of friends who would always be granted access. Our data suggests such a policy would not affect many friends and hence would not enhance an access control mechanism based on communities sufficiently.

When looking for friends who were *always excluded* from the audience, we found a larger variation in the data. All the 31 users had at least one friend who was always excluded from the audience of each of the 10 photos. The average ratio of friends who were never selected was found to be **56% (s.d = 32%)**. This means that the average user excluded more than half of his friend list from each photo. It can easily be envisaged that such friends can be put into a special community which are always denied access by default.

We further examined whether reorganizing the permanently excluded friends into a single community and removing them from their corresponding community would have any positive impact on the penalty calculation. Although we found no significant difference between the algorithms with respect to penalty for exclusion as mentioned earlier, we considered the algorithm with the lowest average penalty, CPM, to recalculate the penalties and observe any changes. This is just to demonstrate that the penalty for exclusion is affected by removing permanently excluded friends, though extent of reduction may depend on the choice of algorithm.

The average reduction in penalty was found to be **59.38% (s.d = 29.5%)**. This can be regarded as a substantial improvement on the penalty values calculated earlier. The high standard deviation in the reduction suggests that the reduction varies for different users. We performed a cluster analysis to try and identify subsets of users based on different levels of penalty reduction using "Two Step Clustering" [27] which generated the following subsets:

1) **High Reduction** - This group of **20** users had an average penalty reduction of **78.18%(s.d = 15.11%)**. The average audience across 10 photos for the average user in this subset was 20.88, the median was 15.93 and the standard deviation was 18.56. Thus, these users can be classified as *Consistently Low Selectors*.
2) **Low Reduction** - This group of **11** users had an average penalty reduction of **25.20%(s.d = 13.57%)**. The average audience across 10 photos for the average user in this subset was 58.25, the median was 45.55 and the standard deviation was 65.28. Thus, these users selected comparatively larger audiences but also had greater variation in their selection as exhibited by the larger standard deviation in audience size.

We also found that the reduction in penalty had a negative correlation with average audience (-0.72), median audience (-0.52) and standard deviation of audience size (-0.80) across 10 photos. This further confirms the notion that highest reduction in penalty can be observed for users who consistently select low audiences and hence have a large number of *always excluded* friends. These users benefit the most from removing the *always excluded* friends from their respective communities.

4.3. Effect of Photo

Source of Photo. Participants chose audiences for 10 photos, 5 randomly chosen from their Facebook account and 5 brought on a USB drive. We observed that there was negligible correlation (all coefficients < 0.1) between the source of the photo and the performance of the algorithms for all the 3 metrics.

Photo Categories. We observed some cases where a clear difference was evident between the categories users selected for the photos which had a very high audience as compared to the low audience photos. For example, one user selected high audience for photos which they categorized as "Event" and "Animals" but low audience for photos which were categorized as "People(friends or family)". This observation prompted us to analyze the effect of photo categories on the audience size of a photo.

As can be seen from Table 3, "People(friend or family)" and "Personal" were the most commonly selected categories. On the other hand, "Celebrity" and "Advertising" were the least selected ones. We also find that "Advertising" and "Celebrity" have the highest average audience. Looking at the average number of audience, it is somewhat surprising to find that Technology has a lower average than presumably more "private" categories like "Personal" and "People(friends or family)".

To give an idea of the privacy implication of each photo category, we examined how often a particular photo from a given category had the minimum and maximum number of audience members for a particular participant. These results are shown in the last two columns of Table 3. We find that "Personal" photos have the highest number of occurrences

TABLE 3: Descriptive statistics of audiences of photos from each photo category

No.	Categories	Photos	Audience Size			No. of times	
			Avg.	S.D	Med.	Max	Min
1	Landscape	33	62.27	69.43	36	5	3
2	People (Others)	24	49.83	68.44	15.5	3	4
3	People (friends)	111	58.95	23.48	28	13	12
4	Architecture	24	48.83	74.28	12	3	2
5	Animals	17	66.35	76.69	33	5	2
6	Travel	79	60.54	71.35	29	12	6
7	Fashion	11	70.09	84.32	23	3	1
8	Celebrity	3	84.33	105.10	42	1	0
9	Event	51	46.55	64.53	19	7	3
10	Humor	31	57.84	74.03	19	4	6
11	Food	19	49.05	70.45	17	3	4
12	Advertising	8	89.75	95	37	2	0
13	Entertainment	36	64.53	75.86	27	3	5
14	Personal	101	54.52	67.21	26	10	15
15	Technology	10	38	55.7	18.5	2	2

when they have the minimum audience for a particular user. However, they also have a substantial number of cases where they have the maximum audience. Similarly, "People (friends or family)" has a high number of minimum as well as maximum audience occurrences. Thus, it can be seen from these results that privacy preferences vary between users and no definitive conclusions can be drawn about privacy implications of categories on their own.

After inspecting this further, we found that there was negligible correlation between the photo categories and the performance of the algorithms according to the evaluation criteria. We also observed that photo categories had negligible correlation with audience size of the photo. We tried to evaluate whether the variability in audience selection of the 11 users for whom the reduction in penalty was low was due to photo categories. We tried a linear regression analysis to try and understand the effect of photo category on audience size for these users but found no significant effect. The correlation coefficients were also low with the highest being 0.287 for "Advertising". Thus, there was negligible or very weak correlation between photo category and audience size even for users selecting variable audiences.

Table 3 also shows that the audience size of most categories had a large standard deviation. While this variation could be attributed to different types of users (high selectors or low selectors) selecting the various categories, we wanted to examine whether categories being paired with each other had any influence in the different average audiences. We did find cases where combination of different categories influenced the average audience. For example, average audience for "Personal" when taken alone is 59.07 whereas when it is paired with "People(others)", it goes up to 92.27. Similar variations can be observed across all categories. Categories which have a high average audience, such as "Advertising" or "Celebrity" can inflate the average audiences for other categories. However, even for pairs or combinations of categories, no conclusions could be made about privacy implications. For example, a particular user selected categories "People (friends or family)" and "Travel" for a particular photo and selected 6 audience members while a different photo shared by a different user with the same categories had 191 members. Such variations were commonly observed across all categories.

5. Discussion

CPM - best overall performance but computationally expensive. Overall, CPM emerges as the best performing algorithm from the findings of our analysis across most metrics and considering most factors. However, the computational complexity of CPM is high (exponential) and other algorithms like Label Propagation (linear) might be options to reduce computational cost but would compromise the performance to some extent. It should also be noted here that communities would not need to be recalculated at the time of making access control decisions, as communities already created statically from the user's friend network would be used at that time. The only time the algorithms would need to be executed to recalculate communities would be when the user's friend network is modified (a new friend is added or a friend is removed).

Even CPM not good enough for everyone. Even the best performing algorithm across all metrics, CPM, produces poor results for the penalty for exclusion metric which seems prohibitive in a dynamic scenario like online social networks. Moreover, the difference between the algorithms for this metric was not significant. We find that CPM has the largest ratio of audience in largest community which means that a large majority of the audience can be created from one CPM community and its difference from other algorithms is significant. However, even if we envisage a scenario where the user selects a particular community entirely and then a few individual users to complete their audience so as to minimize individual exclusions, the performance still seems unsatisfactory. We found that the largest community in the audience (for CPM) contributed about 75% of the total penalty for exclusion for a particular photo on its own. Therefore, even a combination of selecting communities and individuals doesn't enhance audience selection as much as one would have hoped.

Performance can be enhanced for users who consistently select low audience. We observed that penalty for exclusion could be reduced (avg reduction $\sim$60%) by removing "always excluded friends" from their respective communities. This suggests that if access control mechanisms can identify frequently excluded friends over time and rearrange communities to exclude these friends during audience selection, it can produce much better results. These friends can be then put into a single community and a default setting of denying access can be implemented for them. Our results indicate that such a scenario is most effective for users who select consistently low audiences and their penalty for exclusion can be reduced substantially ($\sim 80\%$).

Characteristics of individual and nature of content have no significant effect on penalty for exclusion. We observed that no. of communities per audience and ratio of audience in largest community did have some dependence on personal characteristics of the individual (different algorithms had varying degrees of correlation) while penalty for exclusion

was largely independent of these factors. A noteworthy finding emerging from this analysis is that CPM had the lowest correlation coefficient for most of these factors among all algorithms and can be considered comparatively resilient to these variations in individuals' characteristics as a result. Photo category played a negligible role in determining the size of audience. This suggests that participants interpret categories differently and also that category of the photo alone is not enough to determine audience. We also looked at the performance of algorithms according to the photo categories based on the evaluation metrics but no significant conclusions could be made from that analysis as well.

Limitations. A limitation of the work is that the data and the subsequent analysis is reliant on the activity of the participants during the process. We explained earlier that participants expressed access control decisions for 10 photos, so there was the risk that users might get tired towards the end of the process. However, we found that this was not the case, as there were only 2 users whose audience sizes decreased after the first photos, the minimum average audience per photo for the participants was 10, and many users had audience size of above 50 for the last photo.

6. Conclusion

This paper provides an empirical study of 8 community detection algorithms by checking for a goodness of fit between the communities created by them and the access control decisions made by users during a user study. Overall, the results indicate that community detection algorithms may be a useful tool to create default access control policies for users who have comparatively static access control decisions across the board, e.g., for users who consistently deny access to some friends in their network. For users with more dynamic access control decisions, the community detection algorithms evaluated performed poorly. In terms of particular algorithms, CPM produced the best performance but its computational cost is exponential. A very interesting line of future work would be to study whether users' access control policies can be learned over time and *permanently excluded* members can be removed from their respective communities. As we have shown in this paper, this would have the potential to incrementally improve the goodness of fit between the communities produced and the desired access control decisions of users with more static access control decisions.

References

[1] G. Hull, H. R. Lipford, and C. Latulipe, "Contextual gaps: Privacy issues on facebook," *Ethics and information technology*, 2011.

[2] S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi, "Talking in circles: selective sharing in google+," in *Proc. of the SIGCHI*. ACM, 2012, pp. 1065–1074.

[3] P. Wisniewski, B. P. Knijnenburg, and H. Richter Lipford, "Profiling facebook users privacy behaviors," in *SOUPS2014 Workshop on Privacy Personas and Segmentation*, 2014.

[4] G. P. Cheek and M. Shehab, "Human effects of enhanced privacy management models," *Dependable and Secure Computing, IEEE Transactions on*, vol. 11, no. 2, pp. 142–154, 2014.

[5] S. Jones and E. O'Neill, "Feasibility of structural network clustering for group-based privacy control in social networks," in *Proc. of the SOUPS*. ACM, 2010, p. 9.

[6] G. Danezis, "Inferring privacy policies for social networking services," in *Proc of the 2nd ACM workshop on Security and artificial intelligence*, 2009.

[7] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proc. of the 19th international conference on World wide web*. ACM, 2010, pp. 351–360.

[8] R. L. Fogués, J. M. Such, A. Espinosa, and A. Garcia-Fornes, "Bff: A tool for eliciting tie strength and user communities in social networking services," *Information Systems Frontiers*, pp. 1–13, 2013.

[9] S. Amershi, J. Fogarty, and D. Weld, "Regroup: Interactive machine learning for on-demand group creation in social networks," in *Proc. of the SIGCHI*. ACM, 2012, pp. 21–30.

[10] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks." in *NIPS*, vol. 272, 2012, pp. 548–556.

[11] A. Squicciarini, S. Karumanchi, D. Lin, and N. DeSisto, "Identifying hidden social circles for advanced privacy configuration," *Computers & Security*, 2013.

[12] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[13] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Systems*, 2006.

[14] J. Leskovec and R. Sosič, "SNAP: A general purpose network analysis and graph mining library in C++," http://snap.stanford.edu/snap, Jun. 2014.

[15] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, 2004.

[16] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *ISCIS*. Springer, 2005, pp. 284–293.

[17] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[18] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. of the National Acad Sciences*, 2002.

[19] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[20] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, 2006.

[21] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[22] I. Derényi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical review letters*, vol. 94, no. 16, p. 160202, 2005.

[23] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, 2010.

[24] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[25] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[26] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[27] M.-Y. Shih, J.-W. Jheng, and L.-F. Lai, "A two-step method for clustering mixed categroical and numeric data," *Tamkang Journal of science and Engineering*, vol. 13, no. 1, pp. 11–19, 2010.