The PViz Comprehension Tool for Social Network Privacy Settings

Alessandra Mazzia[*] **Kristen LeFevre**[*] and **Eytan Adar** [*†]
[*]University of Michigan, Computer Science and Engineering, 2260 Hayward Ave. Ann Arbor, MI 48109
[†]University of Michigan, School of Information, 105 South State St. Ann Arbor, MI 48109
{amazzia, klefevre, eadar}@umich.edu

Abstract

Users' mental models of privacy and visibility in social networks often involve natural subgroups, or communities, within their local networks of friends. Such groupings are not always explicit, and existing policy comprehension tools, such as Facebook's *A*udience View, which allows the user to view her profile as it appears to each of her friends, are not naturally aligned with this mental model. In this paper, we introduce *P*Viz, an interface and system which corresponds more directly with the way users model groups and privacy policies applied to their networks. PViz allows the user to understand the visibility of her profile according to natural sub-groupings of friends, and at different levels of granularity. We conducted an extensive user study comparing PViz to current privacy comprehension tools (Facebook's Audience View and Custom Settings page). Despite requiring users to adapt to new ways of exploring their social spaces, our study revealed that PViz was comparable to Audience View for simple tasks, and provided a significant improvement for more complex, group based tasks.

1 Introduction

Online social networking systems have existed for many years, but the changing features of these systems, coupled with mass adoption, have exacerbated the problems of privacy and presentation management. These changes have created a situation in which boundary regulation (Palen and Dourish 2003) is difficult to achieve, and users have difficulty constructing accurate mental models of who can access what. Tools for managing privacy settings in social media frequently couple control (specifying who can access what) with awareness and comprehension (understanding who can access what, given the existing configuration). However, existing tools do not necessarily account for the types of "queries" users would like to make to reconcile their mental models of the system state (or desired state) with the policy defaults of the system, the limitations of the system's privacy management features, and individually-enacted settings.

Currently, sites like Facebook allow users to specify fine-grained policies controlling the visibility of their personal data. For example, Facebook's "Custom Settings" page allows users to specify which pieces of profile data (e.g., *Political Views* or *Status Updates*) are visible to each of their friends. Unfortunately, studies and experience have consistently shown that average users struggle to create, evaluate, and maintain such policies (Acquisti and Gross 2006).

In this paper, we focus on the *policy comprehension* problem. Our goal is to assist the user in understanding the visibility of her data in a natural way. Recent work has observed that users' mental models of privacy and visibility in social networks often involve natural subgroups, or communities, within their local networks of friends (Fang and LeFevre 2010; Jones and O'Neill 2010; Patil and Lai 2005). However, existing policy comprehension tools, such as Facebook's Audience View, which allows the user to view her profile as it appears to each of her friends, are not naturally aligned with this mental model.

In this work, we draw a further distinction between *single tasks*, in which the user seeks to understand whether a data item is visible to a specific friend, and *group tasks*, in which the user seeks to understand whether a data item is visible to a natural subgroup of friends.

Example 1.1. *Consider Margaret, who is evaluating her privacy settings on a popular social network site. Margaret would like to keep in touch with John Self, a high school boyfriend, and former teammates from her high school cross-country team. In Margret's case a single task would be to determine if her phone number is visible to John. Notice that single tasks are easily resolved using the audience view; Margaret can simply view her profile as it appears to John. In contrast, group tasks prove more challenging. For example, determining whether Margaret's phone number is visible to all of her cross-country friends. To answer this question using the audience view requires Margaret to enumerate every member of the cross-country team, and to view her profile as it appears to each of them.*

In a limited set of cases, rule-based interfaces (e.g., Facebook's "Custom Settings" page) can be used for group tasks. However, this typically requires that the user has explicitly constructed a list containing exactly the members of the group (e.g., "Cross-Country Friends"). In many cases, such as when there is no explicit list, or worse, there are conflicting rules (individual friends or lists assigned to both "Make this visible to" and "Hide this from"), the rule-based interface makes group tasks difficult.

To address the policy comprehension problem, we have designed and built a tool, called PViz, which corresponds more directly with users' mental models of privacy. PViz allows the user to understand the visibility of her profile at multiple levels of granularity, and according to natural subgroupings of friends. Section 2 describes the design and implementation of PViz. To support visual exploration we also introduce an initial effort to provide concise, human-readable labels for communities in Section 2.2

We conducted an extensive laboratory-based user study comparing PViz to existing policy comprehension tools (Facebook's Audience View and Custom Settings page). Our results, which are described in Section 3, indicate that PViz and Audience View achieve comparable results for single tasks. For the more complicated group tasks, PViz provides a significant improvement in user accuracy.

2 PViz Overview

The PViz policy comprehension tool is centered on a graphical display, which shows the user's social network. Each node in the display represents a semantically meaningful sub-group of the user's friends (a *community*) or an individual friend. Figure 1(a) shows a screenshot of PViz displaying Margaret's social network. Inspecting the display shows that PViz has found five main communities of friends.

To the left of the graphical display, PViz shows a list of profile items for which the user can configure privacy settings. To view privacy settings for a specific item, the user must select the item from the list. In Figure 1(a), the profile item "Other Phone" is selected.

To interpret privacy settings in PViz the user can observe the color of the node (i.e., community) which ranges from 0% (light) to 100% (dark) and is assigned based on the user's privacy selection for a selected profile item. Alternatively, hoevering the mouse over a node reveals an explicit numerical popup. For example, in Figure 1(a), notice that the node labeled "U. of Alabama" is darker than the node labeled "UGA," indicating that a larger percentage of friends in the "U. of Alabama" community can see Margaret's "Other Phone" than in the "UGA" community.

PViz also includes the ability to view communities and privacy settings at different levels of granularity by zooming in and out. Figure 1(b) shows the process of zooming in on "Brentwood High School," which reveals three constituent sub-communities ("BHS Cross-Country," "Photography Club," and "BHS Soccer"). A hierarchical node-link diagram of this type (e.g., (Perer and Shneiderman 2006; Heer and boyd 2005)) serves the dual purpose of being consistent with both the mental models of "networks" and communities.

In addition to the graphical display, PViz provides several ways of interacting with the social network graph to enhance exploration. For example, the user may search for a friend's name in a search box and the display will automatically center on the node containing that friend. A text box that displays the names of all members of the currently selected node (community).

Example 2.1. *Consider again the single and group tasks from Example 1.1; both are easily completed using PViz.*

To check whether her phone number is visible to John Self (single task), Margaret first selects the profile item "Other Phone," and then uses the search box to find the node containing John. If this node is either black or white, then Margaret knows immediately whether or not John can see her phone number. Otherwise, she must zoom in on the display. At the individual level (Figure 1(c)), notice that the node representing John is white, indicating that John cannot see Margaret's phone number.

To check whether her phone number is visible to her cross country friends (group task), Margaret starts at the coarsest level of granularity, and selects the profile item "Other Phone." She recognizes that her high school cross country friends are a subset of her high school friends, so she zooms in on the node labeled "Brentwood High School." Zooming in reveals a node labeled "BHS Cross Country" (Figure 1(b)). To ensure that the node contains the appropriate friends, Margaret may select the node, and inspect the list of friends who belong to the community. After locating the "BHS Cross Country" node, Margaret can interpret her privacy settings based on the node's color, or hovering the mouse over the node to view the exact percentage.

2.1 Implementation

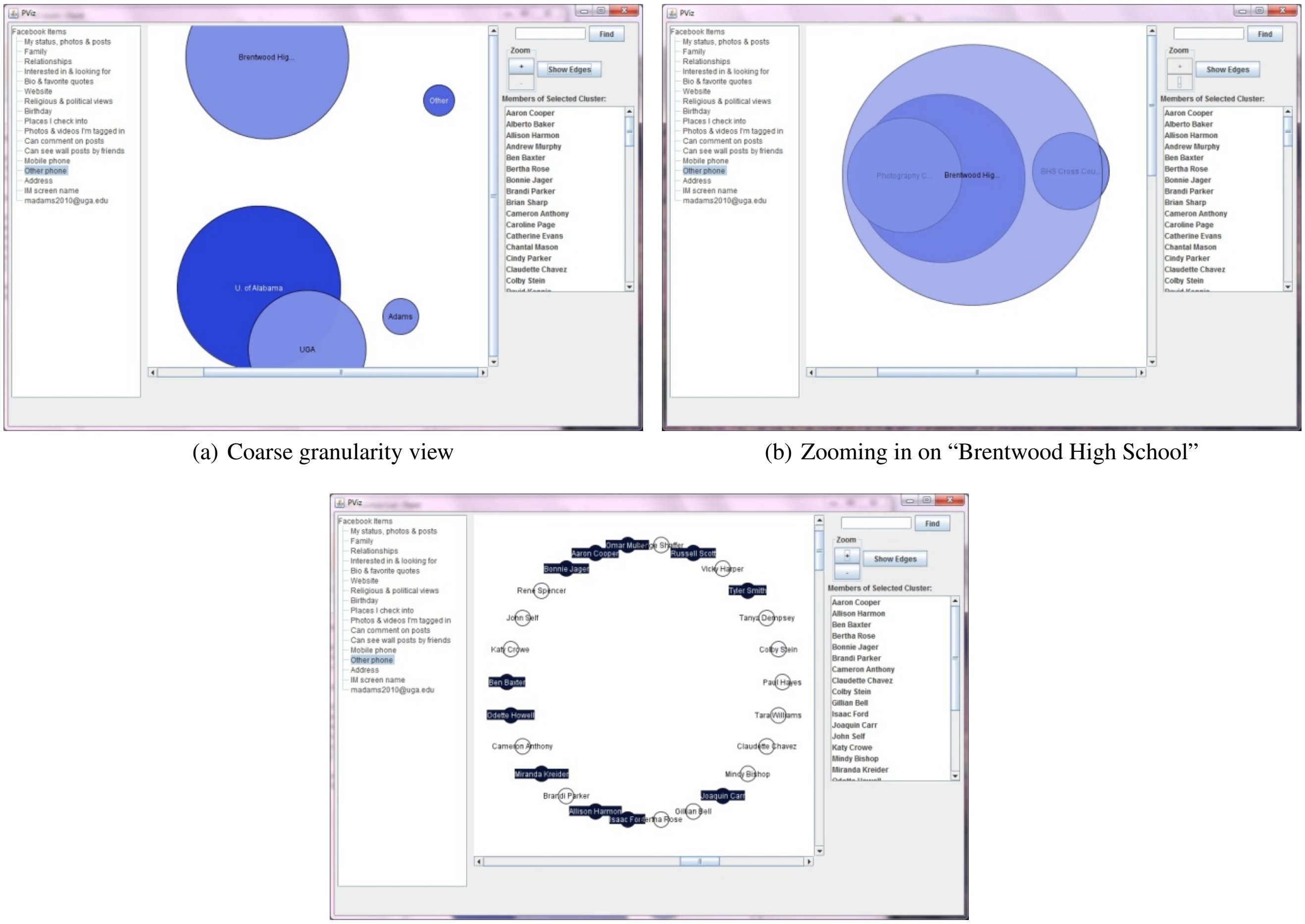
We have implemented a prototype of PViz in the context of Facebook. After the user logs into Facebook, PViz downloads all of the necessary data from the user's account. The current user's friend list, neighborhood network graph (the current user's friends and the friend connections between them), and information from the friends' profiles are all obtained via the Facebook third-party development platform.[1] We have also built a screen-scraping tool to download and process the user's privacy settings, which are not generally available via the open development API.

The problem of partitioning social network graphs into communities has been studied extensively (Fortunato 2010). In PViz, our main goal is not to develop new community-finding algorithms. Currently, we apply a common approach based on the idea of *modularity optimization* (Newman and Girvan 2004; Noack 2009). When finding communities in a social network, it is often difficult to know the right number of communities ahead of time, and modularity provides a natural parameter-free objective function. In the current implementation, we extract a hierarchy of multi-granularity communities according to a simple recursive process in which (1) the network is partitioned into communities based on maximum modularity and (2) each community is treated as another network that is again partitioned. This is repeated until there is no further partitioning that improves modularity. Of course, the PViz interface is general enough that other community-detection algorithms, as well as explicit groupings provided by the user, can easily be integrated.

After the network is partitioned into communities, PViz positions the nodes on the graphical display using a Fruchterman-Reingold (force-based) layout algorithm[2].

[1] http://developers.facebook.com/

[2] As implemented by JUNG, http://jung.sourceforge.net/



(a) Coarse granularity view

(b) Zooming in on "Brentwood High School"

(c) Fine granularity view

Figure 1: PViz allows the user to understand privacy settings at different levels of granularity.

2.2 Keyword Labels

The communities in PViz are labeled using informative keywords. The goal of these labels is to enable the user to quickly identify communities of interest. For example, if a user has a group of friends from the University of Alabama, then presenting a group labeled "U. of Alabama" will help her locate this group. While the user may always configure the labels manually, to save time, PViz generates an initial set of labels automatically.

When choosing labels, we assume that each friend has a set of associated *tags*, which can be compiled automatically from public profile information. Currently, we extract tags from the profiles of Facebook users based on the following fields: Current location (City, State, Country), Home location, High School Name, names of companies listed in Work History, names of universities listed in Education History, affiliated organizations, names of Facebook Groups and "Like" pages.

To support the exploration of the visualization, and to help the user identify the placement of individuals and groups in the visualization space, it is critical to construct informative labels for communities. In designing such a labeling algorithm, we identified two main goals:

1. A community's label should distinguish its members from the rest of the nodes in the graph.
2. Labels should be simple, concise, and easy to understand.

The first goal can be expressed more formally using precision and recall. Let $G = (V, E)$ be a simple unweighted graph with node set V and edge set E. Let $C \subseteq V$ be a community of nodes in G. It is easy to think of a label ℓ as a *query* on the graph, expressed in terms of tags, which returns a subset of nodes $L \subseteq V$. If the query intended to retrieve precisely those nodes in C, then we have $Precision(\ell, C) = \frac{|C \cap L|}{|L|}$ and $Recall(\ell, C) = \frac{|C \cap L|}{|C|}$.

One standard means of combining precision and recall is the F-measure $F(\ell, C) = 2\frac{Precision(\ell,C)*Recall(\ell,C)}{Precision(\ell,C)+Recall(\ell,C)}$.

Definition 1 (F-Measure Labeling)**.** *Given graph G, community C and family of possible labels $\mathcal{L}$, find the label $\ell \in \mathcal{L}$ such that $F(\ell, C)$ is maximized.*

The remaining problem is defining an appropriate *language* for specifying labels in terms of tags. In principle, we could express a label using any logical combination of tags, but this would be complex and difficult for average users to understand. For example, suppose we have three tags: *UGA*, *Tennis*, and *Microsoft*; even if the label

$(UGA \vee Tennis) \wedge (\neg Microsoft)$ uniquely characterizes the members of the community, it is not easy to understand. Thus, we currently restrict the family of possible labels $\mathcal{L}$ to those comprised of a single tag. In this case, labels are easily selected in time linear in the number of tags.

3 PViz User Study

In order to evaluate PViz we conducted a user study comparing it to two alternative tools, Facebook's Custom Settings Page (CS) and Facebook's Audience View (AV), which are representative of the state of the art in comprehension tools for fine-grained social network privacy policies.

We recruited 20 participants (9 women) for the study, all students at our university, with a mean age of 23.3 years. This particular demographic represents a significant fraction of Facebook's user base.[3] In an initial survey, all participants indicated that they had been members of Facebook for at least a year. Self-reported frequency of use ranged from less than once per month to multiple times per day, with most participants indicating that they use Facebook at least once per day. Participants reported a range of experience with Facebook's privacy tools; 70% had previously used the friend list feature, 90% had used the Custom Settings page, and 55% had used the Audience View.

3.1 Standardized Environment

The goal of our study was to compare the utility of PViz to the state of the art policy comprehension tools. An obvious methodology would ask study participants to use each of the three tools to perform single and group tasks related to the visibility of data in their own profiles. Unfortunately, this approach poses several difficult challenges. In particular, in order to evaluate their performance on a comprehensive set of tasks, the participants must have configured their Facebook privacy settings away from the default. According to a recent survey conducted by the Consumer Reports National Research Center, 25% of households with a Facebook account either did not use or were not aware of Facebook's privacy settings.[4] To control for this problem, we instead chose to design an artificial, yet realistic, standardized environment in which to conduct the study.

The standardized environment focused on Margaret, a fictional Facebook user. Her background, social network, friends, profile information, and privacy settings were all created for our study. Margaret had a total of 285 friends (a number consistent with the labeling experiment described below, in which 12 users averaged 297 friends). More importantly, the network was *structurally realistic* as it was based on a real user's network. Margaret kept three Facebook lists of friends: family, graduate school and high school friends. Her privacy settings were configured to allow only a subset of her friends to see each data item. The access control model currently supported by Facebook allows the user to construct both positive ("Make this visible to") and negative ("Hide this from") rules, involving both individual friends and Facebook lists. When configuring Margaret's privacy settings, some data items were given privacy settings that contained conflicting rules, meaning that both positive and negative rules were defined for a specific friend or list.

Rather than creating fake Facebook profiles for Magaret and each of her 285 fictional friends, we created local replicas of Facebook's Audience View and Custom Settings pages. We customized them to reflect Margaret's privacy settings and social network by editing the HTML source downloaded from live versions of the two pages. The local pages mimicked interaction with the online Facebook pages almost exactly, although the peripheral functionality was disabled (e.g., the ability to click on ads).

When completing tasks, study participants were asked to answer from Margaret's perspective. To realistically model Margaret's interaction with the site, we added several additional cues. For example, a group task might ask whether any of Margaret's high school friends can see her status updates. It is easy to identify one's own high school friends, so to mimic this interaction, we annotated the names of Margaret's friends with numeric flags identifying the groups to which they belonged.

3.2 Tasks

We designed 36 tasks to be completed by every study participant. Specifically, we created two categories of tasks:

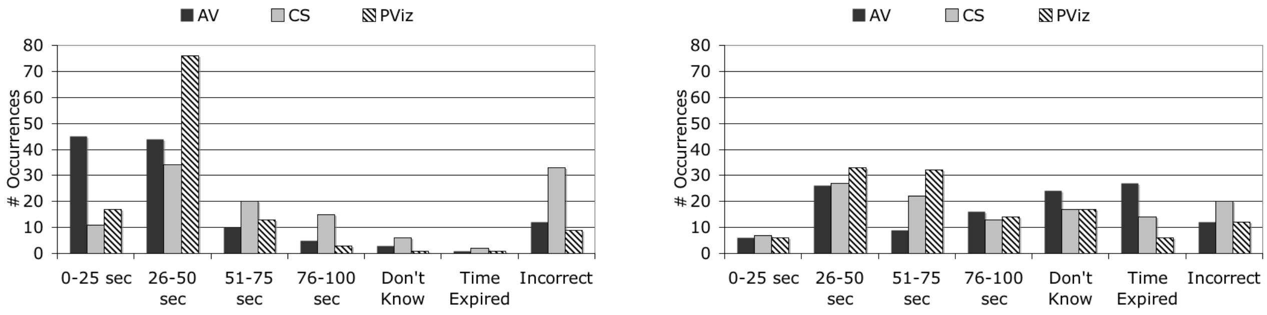
- **Single Tasks** Single tasks ask about the visibility of a data item to a specific friend. (E.g., *Can Alice Smith see Margaret's Date of Birth?*) Half of the single tasks required the participant to resolve conflicting rules on the Custom Settings page, and half did not.
- **Group Tasks** Group tasks ask about the visibility of a data item to a group of friends. (E.g., *Can any of Margaret's high school friends see her Status Updates?* or *What proportion of Margaret's friends from UGA can see her Religious and Political Views?*) Using the Custom Settings page, group tasks are easier if the user has created an explicit list for the given group (e.g., *Family*). Half of the group tasks referred to explicit lists, and half did not. For both types of group tasks, we included some yes/no questions, and also some questions that required the participant to enter a percentage.

For each participant, the tasks were randomly assigned to tools (PViz, AV, and CS). For each tool, each participant was presented with 6 single tasks (3 with conflicts and 3 without) and 6 group tasks (3 with explicit lists and 3 without). The tasks assigned to each tool were then presented in random order. Participants were given a time limit of 1 minute and 40 seconds (100 seconds) per task. Participants had the option of entering an answer for a task, or selecting "I don't know." We measured the amount of time that it took to complete the task, as well as the response accuracy.

Participants completed the study on a desktop computer in a quiet office. Each participant was given detailed background information about Margaret, and presented with each of the three tools in a randomly selected order. For each

[3] 30.8% of users are 18-24 years of age as reported by Facebook's Advertising system on February 4, 2011.

[4] http://www.consumerreports.org/cro/magazine-archive/2010/june/electronics-computers/social-insecurity/overview/index.htm



(a) SingleTasks

(b) Group Tasks

Figure 2: Results summary for single and group tasks. The distribution of times for correctly completed tasks is shown on the left of each chart. The right side of each chart displays the distribution of error cases.

tool, the study administrator explained the functionality of the tool, and walked the participant through a training task. The study concluded with a post-study survey, soliciting participants' thoughts about the three tools.

3.3 Empirical Results

When evaluating a participant's performance on a task, we used two main criteria: (1) Response correctness, and (2) Total time-to-task (measured in seconds). Figures 2(a) and 2(b) summarize our results for single and group tasks, respectively. For tasks completed correctly and within the time limit ($\leq 100s$), the left-hand side of each chart summarizes the distribution of times. Tasks were considered "incorrect," if the participant (1) selected the "I don't know" response, (2) did not respond within the time limit, or (3) provided an incorrect answer.[5] The right-hand side of each chart summarizes the distribution of error cases.

In analyzing the user study data, we first wanted to determine whether the tool (PViz, AV, or CS) significantly affects correctness. For the purpose of this analysis, we coded any task completed correctly and within the time limit as "correct." We coded all other tasks as "incorrect." To account for any serial correlation within participants (since each participant performed multiple tasks), we ran a logistic regression, clustered by participant. The results, which are shown in Figure 3, show that for group tasks, PViz has a significant positive effect on correctness, relative to AV or CS. (The β coefficients for AV and CS are stated relative to PViz. Since both are negative, this indicates that if we were using PViz, but switched to one of the other tools, we would expect the probability of a correctly-completed task to decrease.) For single tasks, PViz has a significant positive effect on correctness relative to CS, but the difference between PViz and AV is not statistically significant. In all cases, we also considered the order in which tasks were presented (e.g., first, second, etc.) to control for the possibility of learning effects; however, such effects were insignificant.

Variable	β	Std. Err.	p
Order	0.0019	0.0173	p = 0.911
Tool=AV	-0.4159	0.5037	p = 0.409
Tool=CS	-1.6027	0.4260	p < 0.001
Constant	2.2623	0.5301	p < 0.001

(a) Single Tasks

Variable	β	Std. Err.	p
Order	0.0045	0.0085	p = 0.597
Tool=AV	-0.9744	0.2520	p < 0.001
Tool=CS	-0.5906	0.2578	p < 0.05
Constant	0.7885	0.3178	p < 0.05

(b) Group Tasks

Figure 3: Results of a logistic regression on correctness, clustered by participant.

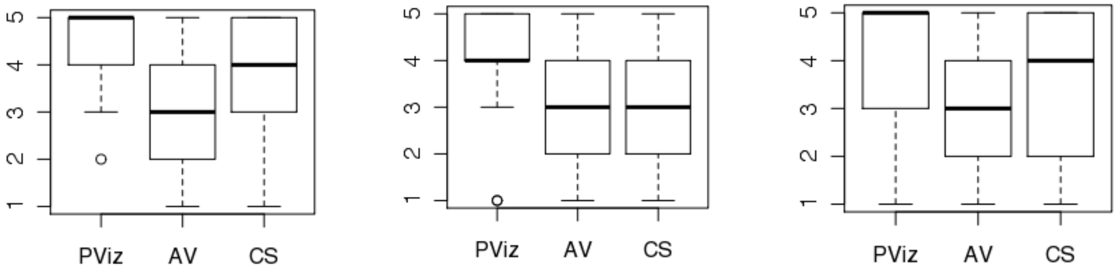
Next, we analyzed the time taken to complete each task. In this analysis, we considered *only* those tasks completed correctly and within the time limit, omitting all others. Figure 4 shows the results of a linear regression on time-to-task, again clustered by participant. For single tasks, we observe that using CS significantly increases the time-to-task, relative to PViz. AV appears to reduce the time-to-task slightly, relative to PViz, but the result is not statistically significant ($p = 0.051$). For single tasks, we also observe a small but statistically significant learning effect. As the value of order increases, time-to-task decreases slightly. For group tasks, neither the tool nor the order has a statistically significant effect on time-to-task.

In the post-survey, we asked participants to assess the tools using three Likert-scale questions. They were asked to respond to each of the following statements on a scale of 1 (strongly disagree) to 5 (strongly agree):

- **Q1:** The tool helped me understand Margaret's privacy settings.
- **Q2:** I enjoyed using the tool.
- **Q3:** I would use the tool on my own Facebook profile.

Figure 5 illustrates the responses to these three questions using boxplots. (The bottom and top of each box indicate the 25th and 75th percentiles, respectively, and the band in the middle indicates the median.) Using a Wilcoxon Signed Rank test (paired by study participant with $p \leq 0.05$), we observed that for question Q1, PViz was rated significantly

[5] For percentage questions, we counted a user's response as correct if it was within 5% of the right answer.



(a) Q1: Helpfulness (b) Q2: Enjoyment (c) Q3: Likely to personally use

Figure 5: Likert-scale responses for user reaction questions

Variable	β	Std. Err.	p
Order	-0.3685	0.1382	$p < 0.05$
Tool=AV	-5.5249	2.6497	$p = 0.051$
Tool=CS	12.875	3.8938	$p < 0.01$
Constant	43.583	3.3622	$p < 0.001$

(a) Single Tasks

Variable	β	Std. Err.	p
Order	-0.2711	0.1758	$p = 0.140$
Tool=AV	-1.7204	4.1824	$p = 0.685$
Tool=CS	-1.2489	3.1696	$p = 0.698$
Constant	60.533	4.8585	$p < 0.001$

(b) Group Tasks

Figure 4: Results of a linear regression on time-to-task, clustered by participant. This analysis only considers tasks completed correctly and within the time limit.

higher than both AV and CS. For question Q2, PViz was rated significantly higher than both AV and CS. For Q3, we observed no significant difference between the three tools.

3.4 Qualitative Feedback

We received a great deal of qualitative feedback from participants. The most frequent comments were suggestions for improvements to PViz navigation (e.g., zooming using the mouse, a button for zooming all the way out, and tools for moving nodes in the display). We plan to incorporate some of these ideas into the next version of PViz.

In general, the qualitative feedback was consistent with our empirical observations. In particular, several participants drew comparisons between PViz and Audience View for single and group tasks:

- *Audience view is useful to check a single friend's view of the profile, but hard to see what an entire group has access to. PViz is much more useable and would make me want to set privacy settings rather than remove information entirely.*
- *Pviz was the easiest to use to get a general idea of who could see what information.*
- *It's very difficult to see percentages on Audience view (you have to check each member of a group to get the %) and settings menu.*

One participant also suggested combining features of the Audience View with the automatically-extracted communities of PViz: *Audience view could present the profile as seen by group. If some members of the group can see different fields it's possible to write the percentages of people that can see this field.*

Participants' reactions to the Custom Settings menu were mostly negative, but one participant did indicate that (s)he had developed a strategy involving a limited number of lists: *I have 3 lists (limited, public, family) and put people in groups according to what I want them to see.*

3.5 Labeling Experiments

To test several community-labeling schemes we recruited 12 additional participants (again, primarily from our university), and used their actual Facebook friends and networks (59-611 friends, mean = 297). While this is not intended to be a representative user sample, it provides an initial comparison of labeling techniques.

We first downloaded each participant's Facebook neighborhood network, including the graph structure, and for each friend, the set of tags described in Section 2.2. We then applied the hierarchical community-detection algorithm described in Section 2.1.

For each community, we showed the participant the list of his / her friends in the community. Then, we asked the question: *On a scale of 1-5, how meaningful is this group (5 = extremely meaningful)?* Finally, we extracted community labels using four alternative techniques:

- **F-Measure:** This labeling algorithm selects the tag that maximizes the F-measure score (see Section 2.2).
- **TF-IDF:** This approach is similar in spirit to the F-Measure approach, but is based on an analogy with the standard IR scoring technique. The idea is to count the number of times a particular tag appears in the given cluster (tf), and to normalize by the log of the number of times the tag appears in the entire network (idf). The algorithm selects the tag with the highest score.
- **Most Common Tag (MCT):** This strawman labeling scheme selects the tag that occurs most frequently among members of the community. Often, labels generated using this approach fail to distinguish members of the community from others in the local network.
- **Logic Rule:** This is another strawman that induces a propositional logic rule, expressed in terms of tags, which distinguishes those friends within the community from those outside of the community. For this experiment, we

used the implementation of the RIPPER algorithm (Cohen 1995) as implemented in the Weka package.[6] Although this algorithm uses aggressive pruning, it often produces more verbose labels than the other techniques.[7]

We then displayed the alternative labels to the user, and asked him or her to select the label that best describes the given community, or to indicate "None of the above."

The participants examined a total of 204 clusters, and selected a label for 53% of these clusters. Figure 6 summarizes the results for the cases where the user selected a label. For each labeling scheme, the y-axis shows the proportion of clusters for which it was selected as best, averaged across users. (The error bars show one standard deviation in either direction.) In some cases, two or more of the labeling algorithms produced the same label, in which case it was counted multiple times. As expected, the TFIDF and F-Measure labels were selected more often than the strawman approaches. (Based on a paired t-test, the difference between F-Measure and Logic is statistically significant ($p \leq 0.05$); the difference between F-Measure and MCT is not significant.) Interestingly, when MCT and Logic did produce good labels, those labels were often also produced by the other algorithms. The right-hand side of the chart describes this phenomenon. For example, the average proportion of clusters for which MCT produced the best label, and that label was *not* also produced by F-Measure, was only 0.18.

Intuitively, one would expect it to be easier to generate labels for "good" clusters. To test this intuition, we considered only the clusters assigned a score $\geq$ the median score awarded by the participant. In this case, participants selected a label for a larger fraction of clusters (64%).

Finally, we were interested in the extent to which we can predict whether a label will be acceptable to the user. We considered only clusters for which the F-measure label was selected or the user specified "None of the above," and we tried to learn a model to distinguish the two.[8] We considered a variety of features (precision of the f-measure label, recall, cluster size, cluster depth, and whether the proposed label is also proposed for another cluster), and ran cross-validation experiments, in which one study subject's data was held out for testing during each trial. We observed average predictive accuracies of 70.5% (C4.5 Decision Tree) and 69.2% (Logistic regression).

4 Related Work

The development of tools to assist average users in specifying, comprehending, and maintaining fine-grained privacy settings is a serious emerging problem in social media. One early study by Acquisti and Gross discovered that while users of social networking sites expressed high levels of concern about their privacy, the same users often did not apply strong privacy policies to their profiles (2006; 2005). In many cases, this was due to users' poor understanding of the available privacy tools and the visibility of their profiles. Broadly, the idea of policy comprehension interfaces has been explored in the HCI community, but with less emphasis on social network systems. For example, Nguyen and Mynatt (2002) offer the idea of Privacy Mirrors as a framework ubiquitous computing infrastructure.

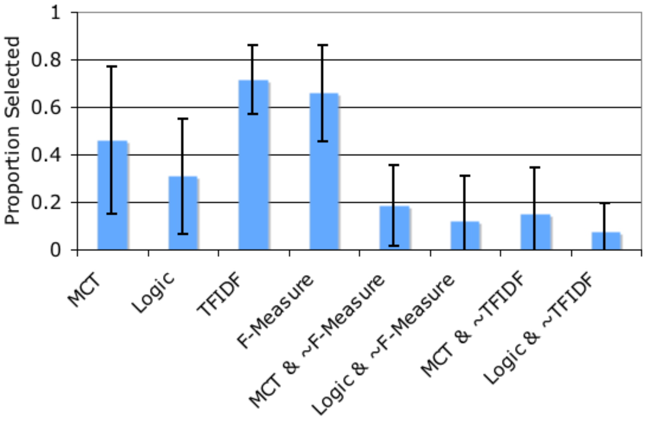


Figure 6: Comparison of Labeling Techniques Based on User Selection

Recent work has sought to address this problem for social networks. Lipford et al. initially proposed and evaluated the Audience View, which allows a user to view her profile as it appears to an individual friend, or as it appears to a manually-specified sub-group of friends (2008). A variation of this interface, which allows the user to view her profile as it appears to an individual friend (no groups), is currently deployed by Facebook.[9]

The expandable grid interface (Reeder et al. 2008) was proposed for the purpose of understanding and authoring access control policies in file systems, but shares several common features with PViz. The expandable grid allows a system administrator to visualize and modify access control settings using a two-dimensional grid–principals (users) × resources (e.g., files)–in which any dimension can be consolidated into coarser groups, or *roles*. Recently, Lipford et al. conducted a pilot study comparing an expandable grid interface with an audience view interface in the context of a social network (2010). While the results did not conclusively favor either of the two interfaces, there are other differences. Perhaps most critically, this study assumed a small set of pre-specified friend groups ("Best Friends," "Family," and "Shady Friends"). In contrast, PViz automatically selects and names meaningful sub-groups, based on social circles that are specific to the individual. This work also points out that compact interfaces (e.g., Expandable grids and PViz) are easier to navigate than the more verbose Audience View when there are many audience groups that are of interest.

Indeed, recent work has sought to understand whether there exist groupings of friends that are natural for the pur-

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] We speculate that this is partially the result of "missing" tags. For example, in Facebook, many of a user's friends might work for $Microsoft$, but only a fraction of these people have indicated this in their profiles. In the future, we imagine that first applying an effective interpolation algorithm might improve the labels induced by the rule learner.

[8] In this data, 57% of examples have the class label "None."

[9] This feature is available by following the "Preview My Profile" link on the Custom Settings page.

pose of controlling privacy. Lampinen et al. document the phenomenon of group co-presence in online social networking sites (2009). Fang and LeFevre (2010) conducted a study in which participants were asked to hand-label their privacy preferences for specific (friend, data item) pairs. They observed that users often expressed homogeneous preferences for friends within the same densely-connected community. Jones and O'Neill (2010) conducted a study in which participants were asked to explicitly group their contacts for the purpose of controlling privacy. They also observed that many users considered structural communities when grouping their friends, in addition to other criteria, such as tie strength. Several others have also advocated the use of structural communities for the purpose of controlling privacy (Adu-Oppong et al. 2008; Danezis 2009).

(Anwar et al. 2009) propose, but do not evaluate, another visualization tool for social network privacy settings. The social network is displayed graphically, and mousing over a node in the graph indicates what that person can access in the current user's profile. Rather than presenting a visualization, (Liu and Terzi 2009) propose computing a single numeric *privacy score*, which communicates to the user the extent to which his privacy settings differ from others' settings. Besmer et al. examine the effects of social navigation cues on users' privacy decisions (2010).

While much work has focused on tools to comprehend and modify privacy settings that already exist, recent work has also proposed using machine learning techniques to recommend privacy settings based on minimal input from the user (Fang and LeFevre 2010).

5 Conclusion and Future Work

In this paper, we introduced the PViz policy comprehension tool for social network privacy. The tool is designed to be more directly aligned with users' mental models of privacy, which often involve natural and user-specific subgroups of friends within their local networks. We conducted an extensive user study comparing PViz to the state of the art. The study indicated that PViz results in significantly better accuracy than existing tools for group tasks and provides support for single tasks that is comparable to the existing Audience View interface.

In designing PViz, our focus so far has been on the privacy comprehension problem (resolving one's mental model of privacy and visibility with the existing system configuration) and hope to provide improvements in this regard (improved community detection and labeling algorithms). However, we believe that PViz also provides a natural platform for policy control. In the future, we plan to extend the PViz tool to include a direct manipulation interface allowing the user to modify her privacy settings.

References

[Acquisti and Gross 2006] Acquisti, A., and Gross, R. 2006. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Privacy Enhancing Technologies Workshop*.

[Adu-Oppong et al. 2008] Adu-Oppong, F.; Gardiner, C.; Kapadia, A.; and Tsang, P. 2008. Socialcircles: Tackling privacy in social networks. In *SOUPS*.

[Anwar et al. 2009] Anwar, M.; Fong, P.; Yang, X.-D.; and Hamilton, H. 2009. Visualizing privacy implications of access control policies in social networks. In *Workshop on Data Privacy Management*.

[Besmer, Watson, and Lipford 2010] Besmer, A.; Watson, J.; and Lipford, H. 2010. The impact of social navigation on privacy policy configuration. In *SOUPS*.

[Cohen 1995] Cohen, W. 1995. Fast effective rule induction. In *ICML*.

[Danezis 2009] Danezis, G. 2009. Inferring privacy policies for social networking services. In *AISec*.

[Fang and LeFevre 2010] Fang, L., and LeFevre, K. 2010. Privacy wizards for social networking sites. In *WWW*.

[Fortunato 2010] Fortunato, S. 2010. Community detection in graphs. *Physics Reports* 486.

[Gross and Acquisti 2005] Gross, R., and Acquisti, A. 2005. Information revelation and privacy in online social networks. In *Workshop on Privacy in the Electronic Society*.

[Heer and boyd 2005] Heer, J., and boyd, d. 2005. Vizster: Visualizing online social networks. *InfoVis*.

[Jones and O'Neill 2010] Jones, S., and O'Neill, E. 2010. Feasibility of structural network clustering for group-based privacy control in social networks. In *SOUPS*.

[Lampinen, Tamminen, and Oulasvirta 2009] Lampinen, A.; Tamminen, S.; and Oulasvirta, A. 2009. All my people right here, right now: Management of group co-presence on a social networking site. In *GROUP*.

[Lipford, Besmer, and Watson 2008] Lipford, H.; Besmer, A.; and Watson, J. 2008. Understanding privacy settings in facebook with an audience view. In *Conference on Usability, Psychology, and Security*.

[Lipford et al. 2010] Lipford, H.; Watson, J.; Whitney, M.; Froiland, K.; and Reeder, R. 2010. Visual vs. compact: A comparison of privacy policy interfaces. In *CHI*.

[Liu and Terzi 2009] Liu, K., and Terzi, E. 2009. A framework for computing the privacy scores of users in online social networks. In *ICDM*.

[Newman and Girvan 2004] Newman, M., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review* 69(2).

[Nguyen and Mynatt 2002] Nguyen, D., and Mynatt, E. 2002. Privacy mirrors: understanding and shaping socio-technical ubiquitous computing systems. Technical report.

[Noack 2009] Noack, A. 2009. Modularity clustering is force-directed layout. *Physical Review* 79(2).

[Palen and Dourish 2003] Palen, L., and Dourish, P. 2003. Unpacking "privacy" for a networked world. In *CHI*.

[Patil and Lai 2005] Patil, S., and Lai, J. 2005. Who gets to know what when: configuring privacy permissions in an awareness application. In *CHI*.

[Perer and Shneiderman 2006] Perer, A., and Shneiderman, B. 2006. Balancing systematic and flexible exploration of social networks. *IEEE Transactions on Visualization and Computer Graphics* 12:693–700.

[Reeder et al. 2008] Reeder, R.; Bauer, L.; Cranor, L.; Reiter, M.; Bacon, K.; How, K.; and Strong, H. 2008. Expandable grids for visualizing and authoring computer security policies. In *CHI*.