Semi-Supervised Policy Recommendation for Online Social Networks

Mohamed Shehab, Hakim Touati
College of Computing and Informatics
University of North Carolina at Charlotte
Charlotte, NC 28233, USA
Email: {mshehab,htouati}@uncc.edu

***Abstract*—Fine grain policy settings in social network sites is becoming a very important requirement for managing user's privacy. Incorrect privacy policy settings can easily lead to leaks in private and personal information. At the same time, being too restrictive would reduce the benefits of online social networks. This is further complicated with the growing adoption of social networks and with the rapid growth in information uploading and sharing. The problem of facilitating policy settings has attracted numerous access control, and human computer interaction researchers. The solutions proposed range from usable interfaces for policy settings to automated policy settings. We propose a fine grained policy recommendation system that is based on an iterative semi-supervised learning approach that uses the social graph propagation properties. Active learning and social graph properties were used to detect the most informative instances to be labeled as training sets. We implemented and tested our approach using real Facebook dataset. We compared our proposed approach to supervised learning and random walk approaches. Our proposed approaches provided high accuracy and precision when compared to the other approaches.**

***Keywords*-Semi-Supervised Learning, Graph-based Propagation, Policy Recommendation, Active Learning.**

I. Introduction

Social networks (SN) contain large amounts of user data. SNs allow users to create and manage accounts, and publish and share information with their friends. Multiple SN sites such as Facebook and Google+ provide fine grained privacy settings. For instance, some SNs allow users to specify a fine grained policy for a certain object at both user and group levels. Posted objects could be as general as a photo album, or as detailed as a comment on a specific photo. With these sophisticated fine grain policy settings, average users are required to manage and administer the privacy of their accounts instead of enjoying the SN services. In addition, the management of privacy policies for a large number of objects and friends is a tedious and complicated task. In response to the SN privacy management limitations, several solutions ranging from visualization tools to policy recommendations were proposed. Pviz is a visualization tool that was proposed by Mazzia et. al [1] to facilitate the privacy settings comprehension. Anwar et. al [2] proposed a visual tool for policy analysis that allows users to visualize their neighborhood access to their data. These tools were limited to help users understand their privacy settings and did not provide recommendations on their privacy settings. Other approaches such as policy wizard [3] and policy manager [4] proposed providing users with policy recommendations based on training supervised learning models. These approaches are based on SN clustering and profile metrics and do not fully exploit SN metrics and network propagation properties. In addition, these approaches focus on building classifiers for a single object and do not generate recommendations for multiple objects posted by the user.

To overcome the limitations of these approaches, we propose a privacy policy recommendation approach based on an iterative semi-supervised learning (SSL) technique. Our proposed approach is based on the clustering assumption that similar or nearby users should have similar labels (permissions). User to user similarity is computed using both the users' profile attributes and SN metrics. Users are asked to label a small set of their friends, ultimately these labels (permissions) are propagated over the SN to provide users with privacy policy recommendations. Furthermore, we propose a mechanism to generate privacy policy recommendations for multiple objects while reducing user effort by leveraging both collaborative active learning and object similarity. The main contributions of this paper are summarized as follows:

- We proposed an SSL based approach to reduce the training set size, and to facilitate label propagation.
- We proposed active learning to reduce user effort by detecting the most informative friends to label.
- We proposed an object collaborative active learning approach to further reduce the user effort.
- We implemented our approach and compared it to supervised learning and random walk approaches.

The rest of the paper is organized as follows: In Section II, we provide a brief background of privacy policies in SNs and semi-supervised learning. In Section III, we present our proposed semi-supervised learning approach, active learning and object collaborative active learning. We present our prototype implementation details and experimental results in Section IV. The related work is presented in Section V. The conclusion is presented in Section VII.

II. Preliminaries

In this section we discuss preliminaries related to notation for modeling social network policies, and graph based semi-supervised learning approaches.

A. Policies in Social Networks

A social network can be represented as a user graph $G_u = (V_u, E_u)$ where V_u represents the set of users and E_u is the set of relationships between users. Each user $u_i \in V_u$ maintains a profile $P_i = \{a_{i,1}, \ldots, a_{i,n}\}$ that is composed of profile attributes $a_{i,k}$ from domain D_k, such as age, location, interests, etc. The user friendship relationships can be represented by edges between the vertices, where a link $(u_i, u_j) \in E_u$ describes a friendship relationship between user u_i and u_j. Users can post content such as photos and can share them with their friends. The user posted objects are also represented by an object graph $G_o = (V_o, E_o)$ where the vertices represent objects and the connections between them represents hierarchical relations between objects, such as parent-child relationships, for example an album and photos in the album.

Most current SNs provide users with customizable fine-grained privacy settings. Depending on the objects' sensitiveness, subjects role, and actions to be performed, SN users define privacy policies. This forces users to perform additional efforts in managing their privacy settings, which becomes too tedious with the large number of friends, objects, and permissions. The privacy policy indicates which subjects (friends) from the user graph G_u who are able to access (eg. read, write) objects from the object network G_o.

B. Semi-Supervised Learning

Semi-Supervised Learning (SSL) is a combination of supervised and unsupervised learning. It uses both labeled data, where the instances' classes are known, and unlabeled data, where only the instance's features are known. SSL uses unsupervised learning to separate the classes' domain regions and uses fewer labeled data than supervised learning to label the domain regions [5]. SSL adopts the consistency assumption, where closer (similar) points have similar classes [6]. Multiple SSL models have been proposed [7], [8], [9], the main differences is in their context of use and the application of the consistency assumption.

Since SNs are mainly represented with graph structures, a graph based SSL is most suited for our task. The assumption behind the graph based SSL is that labels can be propagated from labeled instances to unlabeled instances connected by edges. Instances (nodes) connected with larger edge-weights tend to have same labels, and this naturally translates in SNs as users with stronger connections (larger similarity) have same labels. Another benefit of graph based SSL is to capture the structural topology and contextual information of users. In fact, one of our main assumptions is that users within same clusters have similar permissions. Thus, we propose an iterative approach that uses label propagation on weighted and attributed graphs. For instance, in Figure 1, we show that nodes connected with thicker edges (larger weights) have similar labels (+1, -1). Commonly a graph

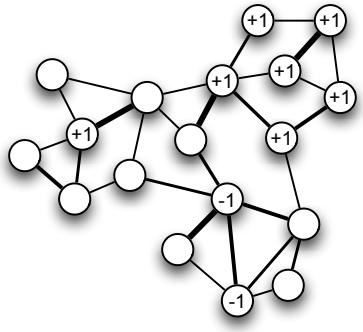


Figure 1. Social Graph Representation.

$G(V, E)$ and edge weight matrix W are used in graph based SSL. The weight matrix W is constructed on the graph G such that each $w_{i,j}$ is the similarity or distance between instances i and j of V. The weight matrix is generally normalized to represent the normalized graph Laplacian as follows: $S = \mathbb{I} - D^{-1/2} W D^{-1/2}$, where D is a diagonal normalization matrix with diagonal elements representing the sum of row of the weight matrix W. Given a graph Laplacian matrix S for a graph G, an initial labels matrix Y, and a diffusion coefficient $\alpha \in (0, 1)$, the SSL propagation function is defined as:

$$F(t+1) = \alpha S F(t) + (1 - \alpha) Y$$

For more details related to SSL the interested reader is referred to [5]. SSL was used previously to exploit information leaks in social networks. However, to the best of our knowledge, this is the first attempt to use SSL in SNs privacy policy recommendations. In the following section, we will describe how we use graph based SSL for SNs' policy recommendations.

III. Proposed Approach

The problem of labeling friends for privacy policy settings can be modeled as a two-class classification problem, where the class labels include the policy actions Allow and Deny. We strive to predict privacy label recommendations from previous user preferences while maintaining minimum user involvement. To minimize the number of labeled data, thus the user's effort, we use an iterative semi-supervised learning approach. The main steps of our approach are described in the Figure 2.

Given an attributed weighted graph $G(V, E)$ representing a SN graph, we construct the weighted adjacency matrix W. The similarity computation is important for SSL based approaches, since it guides the label propagation. The similarity between users is computed using their feature vectors. In online SNs there is a large number of directly available features (such as profile attributes and social graph), in

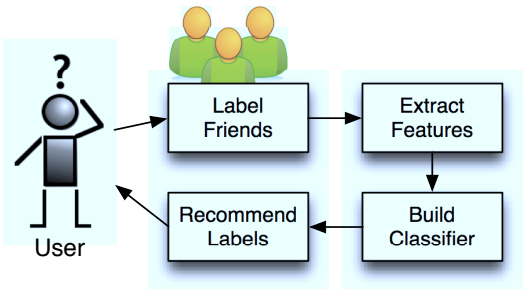


Figure 2. Policy Learning Model

addition there are features that can be extracted from the SN graph (such as network metrics and spectral coordinates). For each user the profile information, network metrics and community information are described using the profile vector X_P, network vector X_N, and spectral coordinates X_S respectively. The network represents the importance and status of the user in the SN using metrics such as centrality, betweenness and degree [10], [11], [4]. The spectral coordinates are computed based on the eigenvectors of adjacency matrix of G. There are multiple approaches for merging the user's contextual and spatial features, some of which are described in [9]. In this paper, we adopt both the simple stacking and weighted approaches. The simple stacking approach simply combines all the user's feature vectors into a single vector $X = (X_P, X_N, X_S)$. The similarity $w_{i,j}$ between users i and j feature vectors X_i and X_j using the RBF kernel. The RBF kernel is used for its proven effectiveness and its simplicity having only a single bandwidth coefficient σ. We also used other similarity techniques to compare against the RBF kernel, such as the cosine similarity; however the RBF kernel gave the best results. The similarity between users i, and j is set to 0 if $(i,j) \notin E$, otherwise the value of $w_{i,j}$ in the similarity matrix is computed as follows:

$$w_{i,j} = e^{(-\|X_i - X_j\|^2/2\sigma^2)} \; \forall i \neq j$$

The weighted approach generates a weight sum of similarities computed using each feature vector separately. Through this approach we are able to control the contribution of each feature vector in the computed similarity. The weighted sum is computed as $w_{i,j} = \beta_1 \; w_{i,j}^{X_P} + \beta_2 \; w_{i,j}^{X_N} + \beta_3 \; w_{i,j}^{X_S}$, where $\beta_1 + \beta_2 + \beta_3 = 1$ and $w_{i,j}^{X_P}$ represents the similarity computed based on X_P. Both the stacking and weighted approaches generated similar results due to the use of Principal Component Analysis.

To select the initial set of friends to be labeled is an essential step that has an effect on the classification process. We are eager to minimize the number of initial labeled friends to reduce the user's effort. We leverage the Clasuet Newman Moore (CNM) network clustering algorithm [12] to cluster the user's SN graph into clusters of friends. From each cluster the nodes with high betweenness centrality values are selected to be labeled by the user, this ensures that the selected nodes have a contribution to the SSL label propagation [13]. In addition, the initial training set was chosen by community and permission type. This allows us to build an initial labeled dataset that represents each cluster and to balance the permission types. In other words, users label an initial balanced number of allowed and denied friends for each computed cluster.

We represent the initial labeled data with an $n \times 2$ binary matrix Y representing the user's permissions for a particular object $o \in O$, where n is the number of users and each $y_{i,j}$ represents the user i permission label represented by the column index j (where $j = 0$ represents the Deny column and $j = 1$ represents Allow), where (1,0), (0,1) and (0,0) represent deny, allow and unlabeled respectively. For example, Bob posted a new object and would like to setup his privacy policy for his friends David, Alice and Marry. He denies access to David, allows access to Alice and Marry is left unlabeled. The initial matrix Y is as follows:

$$\begin{array}{c} \\ David \\ Alice \\ Marry \end{array} \begin{array}{c} \begin{array}{cc} Deny & Allow \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \end{array}$$

A. Label Propagation

For the classification of the users' permissions, we use a semi-supervised propagation function in an iterative process. In this process, we propagate the users' permissions throughout the similarity graph. At each step, the labeled data along with their similarity to each user are used to propagate the policy labels. The classification function is described as $F(t+1) = \alpha S F(t) + (1-\alpha)Y$, where Y, S and $F(t)$ represent the initial labels matrix , similarity matrix, and previous iterations matrix respectively. The iterative function $F(t)$ converges to F^*, as proven in [9], described as follows:

$$\lim_{t \to \infty} F(t) = (1-\alpha)(\mathbb{I} - \alpha S)^{-1} Y.$$

The closed form noted F^* will result in a $n \times 2$ matrix, where the first column represent the *Deny* preference and the second column represents the *Allow* preference. A permission recommendation decision is made for each friend by choosing the highest preference value. This iterative propagation process depends on the initial labeling and the accuracy can be improved by using the active learning paradigm [14]. The propose active learning approach is discussed in the following section.

B. Active Learning

Active learning [15], [16] aims at reducing the number of training examples to be labeled by selectively picking a subset of the unlabeled data. The selection of data points to label is done by inspecting the unlabeled examples and selecting the points with the maximum label ambiguity

(entropy), or least confidence. In other words, active learning aims at selecting the examples which have the largest improvement on performance and ultimately reducing the amount of human labeling effort involved.

Let the set of instances $X = X_l \cup X_u$, where $X_l = \{x_1, \ldots, x_l\}$ is the set of labeled instances and $X_u = \{x_{l+1}, \ldots, x_n\}$ is the set of unlabeled instances. Active learning will sequentially select the most informative (uncertain) friend instance u_k from the unlabeled pool X_u to be labeled and added to the labeled pool X_l. The newly labeled user u_k is the user with the least confidence $C_k(i)$ among the unlabeled friends in the previous iteration. The proposed confidence vector $C(i)$ for the unlabeled friends is computed using using the previous iteration $F^*(i)$ as follows:

$$C(i) = \frac{\|F^*(i)[Allow] - F^*(i)[Deny]\|}{\|F^*(i)[Allow] + F^*(i)[Deny]\|}$$

Where $F^*(i)[Allow]$ represents the $allow$ preference column with the index one, and $F^*(i)[Deny]$ represents the $deny$ preference column with the index zero. Assume that initial label assignment is represented by Y_i. After selecting the least confident instance k, the new label set is denoted by Y_{i+1} such that $Y_{i+1} = Y_i + \Delta_{i+1}$, where $\Delta_{i+1} \in R^{n \times 2}$ represents the labeling of the newly labeled instance and is zero else where. If the instance $x_k \in X_u$ is selected for labeling, where the user is asked to label for his friend u_k, the value of Y_{i+1} is:

$$Y_{i+1} = Y_i + \begin{bmatrix} 0 & \cdots & y_{1,k} & 0 & \cdots \\ 0 & \cdots & y_{2,k} & 0 & \cdots \end{bmatrix}^T$$

Where $y_{1,k}, y_{2,k} \in \{0, 1\}$ are the allow and deny choices respectively for user's friend u_k. Given the new labeled matrix Y_{i+1} the final labeling can be computed based on the label propagation approach to compute $F^*(i+1)$. In what follows we show that the new value of $F^*(i+1)$ is related to previous $F^*(i)$, which can be computed incrementally as:

$$\begin{aligned} F^*(i+1) &= (1-\alpha)(I - \alpha S)^{-1} Y_{i+1} \\ &= F^*(i) + (1-\alpha)(I - \alpha S)^{-1} \Delta_{i+1} \end{aligned}$$

Note that the second term is simply the addition of $[0, A_k]$ or $[A_k, 0]$, which is based on the value of $y_{1,k}$ and $y_{2,k}$, to the previous $F^*(i)$, where A_k is the k^{th} column of the matrix $A = (1-\alpha)(I-\alpha S)^{-1}$. The added term summarizes the propagation contribution of the newly added labeled user u_k. The matrix A is a constant matrix computed once at the beginning of the process. Therefore, at each active learning iteration the new labeling $F^*(i+1)$ is simply the sum of $F^*(i)$ and the propagation contribution, without the overhead of matrix multiplication or inverse.

C. Object Collaborative Active Learning

To reduce users labeling effort, we propose an object collaborative active learning (CAL) technique that leverages object similarity in the active learning approach. The proposed CAL approach starts by grouping similar objects together, then it computes a global confidence vector C_g for each group g. The global confidence vector C_g is used to pick the least confident friend to be labeled for each object in the group.

Algorithm 1: Object CAL

Input: Users X, Objects O
Get initial labelings Y_o for each object $o \in O$.
Cluster objects based on similarity into groups (OG).
repeat
  for $g \in OG$ **do**
    Compute $F_o^*(i)$ for each object in object group O_g.
    Compute confidence $C_g(i)$ for each object group O_g.
    Based on the confidence $C_g(i)$ select the unlabeled user u_k that has the least confidence.
    Request the user to label the selected user u_k.
    Update the labeling set Y_o for each object in the object group O_g to reflect the labeling of u_k.
  end
until *User continues to provide privacy labels*;

We group objects using the hierarchical structure of the object network when possible, for instance photos in the same album. In addition we group similar objects with regards to objects' initial policies using a simple similarity clustering algorithm, such as k-means. We assume that similar objects, with regards to users' initial policy preferences will have similar policies and thus can collectively guide the active learning process to select the next user to label. The SSL propagation technique will lead to similar policy preference predictions for objects with similar initial permission labels (policy vectors). This will result into similar confidence values and similar friends to label in the CAL, thus reducing the labeling effort.

Global confidence for each object group O_g is computed using the confidence vectors of each object in the group. Recall that the unlabeled friends selected for labeling in the active learning are the points with maximum preference ambiguity or least confident for each object. Assume $C_o(i)$ is the friends' confidence vector computed using the propagation function $F_o^*(i)$ for each object o, a combined confidence vector $C_g(i)$ can be computed for all objects $o \in O_g$ in each CAL group, such that:

$$C_g(i) = \sum_{o \in O_g} \delta \; C_o(i).$$

Where $\delta \in (0, 1)$ is the object weight or object sensitivity coefficient that is set by the user following the object importance. Once $C_g(i)$ is computed for a specific group, we choose the friend with the least confidence among the users of $C_g(i)$, and ask the user to label this friend for each object in the group. All objects of a specific group contribute to picking a common friend to be labeled for each object of the group, and thus reduce the effort required in comparison to setting the privacy policy for each object separately.

IV. Experiments and Evaluation

In this section we start by describing the prototype implementation details, then we discuss the evaluation results comparing our proposed approach with other approaches.

A. Prototype Details

To collect user policy preferences we created a user study as a Facebook application for profile policy management. The application provides users with an interface to specify policies for their friends. The application user interface was designed to be aesthetically similar to the current Facebook profile screen. The profile was made to look as real as possible by including the current participant's profile information however the provided friends' permissions are only for the purpose of the study. The application presents the user with one of his friends and is asked to decide on what profile attributes this friend should be allowed/denied access to. The interface was built such that user is able to see what his friend will be able to view, by clicking an attribute the user is able to toggle the access granted to his friend over this attribute. In addition the attribute alpha level is changed to indicate if it is allowed (bold) or denied (grayed out). Figure 3, shows a screen shot of the prototype, for example the friend is allowed to access the user's religious views, and is denied access to the user's email.

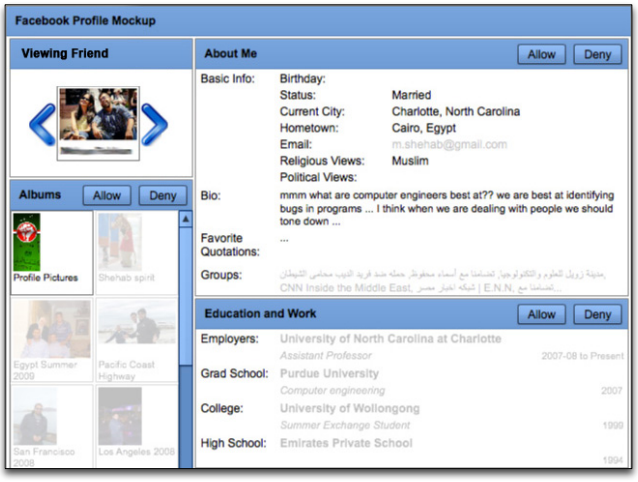


Figure 3. Policy Authoring Prototype.

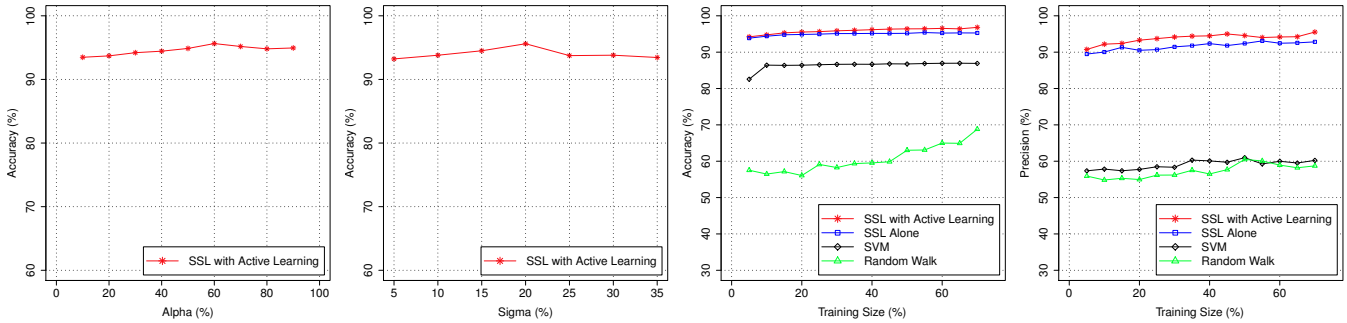
The user study, which was approved IRB Protocol #11-08-01 UNC Charlotte, was composed of two tasks. In the first task the participants installed our Facebook application which collects user profile information, friendship relationships (social graph), friends' profiles, and group information. Then the users were presented with an online tutorial highlighting how to application's user interface to compose profile policies. In the second task, the users were presented with our interface and were asked to provide profile policies for their friends. We implemented our prototype using Adobe Flex 4 for the client side and PHP for Facebook APIs access. We collected users access permissions on email, birthday, college, status, location, religion, photos, albums, and more.

We recruited our user study participants from the student population of the University of North Carolina at Charlotte and from Amazon Mechanical Turk. Amazon Mechanical Turk is a crowd sourcing marketplace that pairs *Requesters* of work and *Workers*. Requesters formulate work into Human Intelligent Tasks (HIT) which are individual tasks that workers complete. We set up our prototype Facebook application as a HIT, which included the two tasks described. To better control the quality of the recruited participants, we mandated that each worker have at least a 95% HIT approval rating. A HIT took approximately 10-15 minutes to complete, for which each worker was paid a fee of $1.50. A 222 users successfully completed the user study. We used the total time spent to complete the study as a measure to remove 5% of the outlying users who had an absolute Z-Score value greater than three.

B. Experimental Results

The SSL based approach and the proposed active learning were implemented using the statistical *R* language version 2.14.0. We also implemented the supervised learning approach described in [3], and the random walk approach described in [14] to compare against our approach. The supervised learning approach as the name implies uses a supervised learning classifier (SVM) to classify and label users. The random walk approach constructs an active learning approach based on random walk on graphs and models the transition probabilities proportional to the graph edge weights. This approach uses the conditional expected hitting times from unlabeled nodes to labeled nodes as a classification rule. The experiments were conducted on the collected privacy policy preferences. We used Facebook APIs to extract users' profile information, friends (social graph), and objects information belonging to users. To extract network metrics, we used the social graph topology and we computed the betweenness, degree, and centrality. In addition, we computed the friend's spectral coordinates from the adjacency matrix using the eigenvectors. We conducted a number of experiments starting by testing the SSL approach. We first tested the discriminative attributes against the training effort and divided the attributes into three classes (Profile, Network, and Spectral attributes). We noted that by combining all the three classes into a single feature vector consistently gave higher accuracy and precision values, thus in all the following experiments we used all the three feature classes.

Parameter Selection: the SSL approach uses a diffusion coefficient α that guides the bias either towards the initial training Y provided by the user or towards the similarity propagation. We conducted an experiment to find the best



(a) Accuracy Varying Alpha (b) Precision Varying Sigma (c) Accuracy for All Approaches (d) Precision for All Approaches

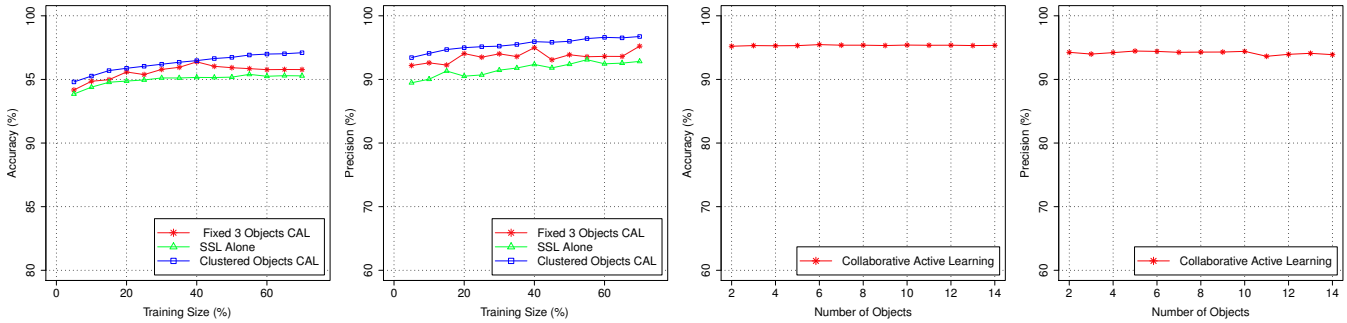
Figure 4. Experimental results: parameter selection and classifier comparison

value of this parameter, Figure 4(a) shows that setting the diffusion coefficient to 60% provides the best accuracy result. In addition, we performed an experiment to find the suitable bandwidth parameter σ used in the RBF kernel similarity computation. Figure 4(b) shows that the bandwidth parameter of 20% returns the best accuracy. In all the following experiments we used 20% as bandwidth parameter and 60% as diffusion coefficient. We also ran 10 folds cross-validation for the training in each experiment.

Classifier Comparison: We conducted experiments to evaluate our graph-based SSL approach against the training effort of the users and to compare it to the existing Supervised and Random Walk approaches. Figures 4(c) and 4(d) show that our approach performs at over 94% accuracy and 91% precision with only 5% training effort, and increases monotonously with the user training effort. In addition, our proposed SSL approach outperforms both the supervised learning and random walk approaches by providing higher accuracy and precision for the same effort levels. Note that the random walk approach provided accuracy and precision levels lower than both SSL and supervised learning (SVM). The supervised learning approach failed to provide high precision which averaged around 60%, while on the other hand our proposed approaches maintained a precision over 90%. We also investigated the effect of our proposed graph based SSL with active learning allowing users to label one node at each active learning iteration, where we used 20% of the training effort as initial training and 80% as active learning training effort. Figure 4(c), shows that in comparison to the SSL approach the SSL with Active Learning provides about 2% improvement to the accuracy and 3% improvement to precision at the same training effort.

Collaborative Active Learning: To evaluate the CAL approach, we select three objects and attempt to train them simultaneously using the proposed approach. Even though CAL uses the same friends to label as active training set for all the three objects, thus reducing the user's effort, there is no reduction in the classification accuracy and precision when compared to SSL Alone as shown in Figures 5(a) and 5(b). The values reported for SSL Alone, represent the average values computed for the three trained objects. In addition CAL provides around 4% improvement in accuracy and precision when compared to SSL Alone. This implies that the CAL is able to simultaneously train multiple classifiers while at the same time reduce the user effort. We also used a clustering technique to cluster similar objects together into groups and use each group in the CAL. We chose a pool of 14 objects for this experiments, and the average number of clusters was of 1.3 clusters with an average of 10.8 objects per cluster. Regardless of the number of objects in each group, our experiments show that using similar objects of a group in CAL improves the average accuracy and precision in comparison with the SSL Alone and the randomly selected three objects as shown in Figures 5(a) and 5(b). Since we are using the user training effort (80% of the training size) for all three objects, we can lower the user effort by 66%. Given n objects used in the CAL with initial active learning training effort e (number of friends to label), the user effort is reduced to $\frac{e}{n}$. To investigate the effect of varying the number of objects, we fixed the initial training effort to 10% and varied the number of objects selected randomly with multiple folds. Increasing the number of objects in CAL decreases the user's effort, while maintaining the accuracy and precision values. Figures 5(c) and 5(d), report the accuracy and precision of the CAL approach for different number of objects.

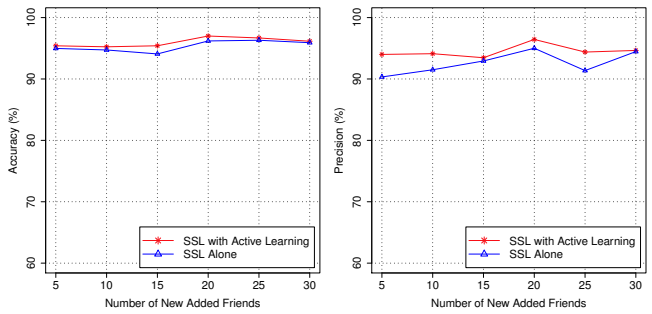
Incremental Study: SNs are dynamic environments where friends and objects are continuously being updated. Users are always making new friendships, which translates to adding new links to the social graph, in addition users can also break their friendships with other users. Similarly, users can add and delete objects such as photos and videos. This incremental behavior leads to increase the number of privacy policy settings, thus the number of users' decisions. We performed an incremental study on the user population that we have collected by selecting a subset of the SN (social



(a) Accuracy for CAL by Effort (b) Precision for CAL by Effort (c) Accuracy for CAL by Object (d) Precision for CAL by Object

Figure 5. Collaborative Active Learning (CAL) Experimental Results.

subgraph) of each user and incrementally adding friends. This approach consist of rebuilding the propagation function with a new similarity matrix based on the new social graph topology and clustering. Figures 6(a) and 6(b), show the effect of adding new friends on the accuracy and precision for both the SSL Alone and SSL with Active Learning mechanisms. Note that the classifier is able to provide high accuracy and precision even with adding new nodes (friends) to the social graph.



(a) Incremental Study Accuracy (b) Incremental Study Precision

Figure 6. Incremental Study Results

V. Related Work

Privacy settings in Social Networking sites is a big concern for users; in addition, fine grained privacy settings makes it harder for users to specify their privacy preference. Access control in social networking sites is an emerging problem that is attracting numerous access control researchers providing assisted privacy setting tools, privacy wizards, and recommendation tools. This paper mainly focuses on privacy preference recommendation, and builds upon our previous work in user centric policy management [4].

In our previous work [4], we proposed a supervised learning approach to build a classifier model based on interaction, which evaluates trust between users. We also fused the different community classifiers to improve the classification results. Although the results were promising (83% accuracy, 78%precision with 20% training), this approach did not address active learning and multiple object labeling. Another related work is Fang et al., they proposed a privacy wizard using supervised learning that uses community structure and proved that the friendship based community features can provide accurate classification results [3]. Fang et al. used uncertainty sampling to identify the users most uncertain friends to label. In this paper, we use a semi-supervised learning approach that improves on the previously used supervised learning approach using less labeled data. In addition, we use a novel clustering approach to select the most informative friends based on friends feature vectors, adjacency matrix, and policy labels.

In this paper we use a semi-supervised learning technique for privacy preference recommendations. Numerous work have been done to demonstrate the effectiveness and efficiency of semi-supervised learning in comparison to supervised learning. For instance, semi-supervised learning was used to optimize feature selection in graph classification in [17], where the authors extracted optimal subgraph features using labeled and unlabeled graphs and proved that their approach outperformed the supervised and unsupervised approaches. One of the most important elements of semi-supervised learning is the selection of the type and algorithm to use following the domain of interest. There is several semi-supervised learning techniques including: Semi-supervised learning using Co-Training [18] where the idea is to use less labeled data by dividing the classification problem into two separate views and train two models recursively using each other training data. Graph based semi-supervised learning is another approach that models the problem as a weighted graph problem, where the vertices represent the labeled and unlabeled data and the weighted edges represent a similarity between the nodes [19]. A comparative study of semi-supervised learning techniques can be found in [20].

More relevant to our work, semi-supervised learning was

used in social networks to infer users private information from the public labeled and unlabeled data using co-training and graph based semi-supervised learning [21]. Semi-supervised learning was also used in Hyper-spectral Image Classification [9], where the authors used a graph based semi-supervised learning by recursively improving upon previous predictions. The authors also used a closed form that was proven to converge in [22].

Random walk is another approach used in class prediction. For instance, it was used in Active Learning via Random Walk [14], where the authors used a relative distance based on random walk probability to predict the most informative instances to label. Interesting fact about this paper is that the authors chose the closest points to both classes and still used a relative distance to handle outliers.

Collaborative filtering is popular in recommender systems and we adopted this principle in policy recommendations. It was used in [23] to predict user preferences by implicitly extracting users ratings and using user based and item based collaborative filtering. Fouss et al. [24] proposed a collaborative filtering model based on random walk on a graph and extracted two quantity, average first passage time and average commute time, then embed those quantities in an Euclidian space to identify the recommendations. A survey of collaborative techniques can be found in [25].

VI. Acknowledgements

This research was partially supported by grants from the National Science Foundation (NSF-CNS-0831360, NSF-CNS- 1117411) and a Google Research Award.

VII. Conclusion

Policy settings in social networks is a challenging problem for SN users, and at the same time automation approaches create multiple issues. The paper presented a solution for helping users to manage and set their privacy policies through policy recommendations. We proposed a graph based SSL for policy recommendation in SNs to reduce user effort and improve the policy preference prediction accuracy. Furthermore, we proposed an active learning and collaborative active learning approach to reduce the user effort while providing high prediction accuracy and precision. We implemented the proposed approaches in the context of a real world social network (Facebook) as a social application. The collected data was used to conduct experiments on the proposed approaches. The experimental results highlight the effectiveness of the proposed approaches in comparison to supervised learning and random walk approaches. In addition, we showed the ability of the proposed SSL approach to accommodate the dynamic behavior of SNs.

References

[1] A. Mazzia, K. LeFevre, and E. Adar, "The PViz Comprehension Tool for Social Network Privacy Settings," University of Michigan, Tech. Rep. CSE-TR-570-11, April 2011.

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

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

[4] M. Shehab, G. Cheek, H. Touati, A. C. Squicciarini, and P.-C. Cheng, "User centric policy management in online social networks," in *Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks*, 2010, pp. 9–13.

[5] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool, 2009, pp. 9–40.

[6] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," in *The MIT Press*. Massachusetts Institute of Technology, 2006.

[7] J. Ratsaby and S. Venkatesh, "Learning from a mixture of labeled and unlabeled examples with parametric side information," in *Annual Conference on Computational Learning Theory*, 1995.

[8] T. Joachims, "Transductive inference for text classification using support vector machines." in *16th International Conf. on Machine Learning*. Morgan Kaufmann, 1999.

[9] G. Camps-valls, S. Member, T. V. B, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 2044–3054, 2007.

[10] M. E. J. Newman, "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality," *Physical Review E*, vol. 64, no. 1, pp. 016 132+, June 2001.

[11] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, October 2006.

[12] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, pp. 1– 6, 2004.

[13] Prasad Balkundi and Martin Kilduff , "The ties that lead: A social network approach to leadership," *Elsevier Inc.*, 2006.

[14] A. Mukherjee and J. Chen, "Active learning via random walk," 2010. [Online]. Available: http://www.eecs.umich.edu/~cscott/past_courses/eecs545f09/projects/ChengMukherjee.pdf

[15] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[16] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, pp. 201–221, May 1994.

[17] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.

[18] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

[19] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *IN ICML*, 2003.

[20] A. A. Benczr, K. Csalogny, L. Lukcs, and D. Siklsi, "Semi-supervised learning: A comparative study for web spam and telephone user churn," in *In Graph Labeling Workshop in conjunction with ECML/PKDD*, 2007.

[21] M. Mo, D. Wang, B. Li, D. Hong, and I. King, "Exploit of online social networks with semi-supervised learning," *Neural Networks (IJCNN), The 2010 International Joint Conference*, 2010.

[22] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 321–328.

[23] M. Papagelis and D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," *Engineering Applications of Artificial Intelligence*, vol. 18, 2005.

[24] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.

[25] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, vol. 2009, January 2009.