2012 IEEE 12th International Conference on Data Mining

Risks of Friendships on Social Networks

Cuneyt Gurcan Akcora, Barbara Carminati, Elena Ferrari
DISTA, Università degli Studi dell'Insubria
Via Mazzini 5, Varese, Italy
{cuneyt.akcora, elena.ferrari, barbara.carminati}@uninsubria.it

***Abstract*—In this paper, we explore the risks of friends in social networks caused by their friendship patterns, by using real life social network data and starting from a previously defined risk model. Particularly, we observe that risks of friendships can be mined by analyzing users' attitude towards friends of friends. This allows us to give new insights into friendship and risk dynamics on social networks.**

I. Introduction

Users register on social networks to keep in touch with friends, as well as to meet with new people. Research works have shown that a big majority of people that we meet online and add as friends are not random social network users; these people are introduced into our social graph by friends [2]. Although friends can enrich the social graph of users, they can also be a source of privacy risk, because a new relationship always implies the release of some personal information to the new friend as well as to friends of the new friend, which are *strangers* for the user. This problem is aggravated by the fact that users can reference resources of other users in their social graph; and make it very difficult to control the resources published by a user. This uncontrolled information flow highlights the fact that creating a new relationship might expose users to some privacy risks.

We cannot assume that friends will make the right choices about friendships, because friends may have a different view on people they want to be friends with. Considering this, privacy of a social network user should be protected by building a model that observes friendship choices of friends, and assigns a risk label to friends accordingly. Such a model requires knowing a user's perception on the risks of friends of friends. We made a first effort in this direction in [3] by proposing a risk model to learn risk labels of strangers by considering several dimensions. To validate the model, we developed a browser extension showing for each stranger (i) his/her profile features, (ii) his/her privacy settings, and (iii) mutual friends. Based on this information, the user is asked to give a risk label $l \in \{1, 2, 3\}$ to the stranger. These risk labels correspond to *not risky, risky* and *very risky* classification of a stranger. Through the extension, 47 users (32 male, 15 female users) have labeled 4013 strangers. However, we did not consider *risk of friends*.

This new work starts with considering two factors in assigned risk labels. First, strangers can be risky only because of their profile features. Second, a friend himself can increase or decrease the risk of a stranger. Increases and decreases will be termed as *negative and positive friend impacts*, respectively. In any case, if a risky stranger is introduced into the user's social graph it is because of his/her friendship with a friend. However, determining the friend impacts can help us to determine which privacy actions should be taken to avoid data disclosure. We aim at learning how risk labels are assigned to strangers depending only on their profile features, and how much a friend can impact (i.e., increase or decrease) these labels. If strangers are risky just because of their profile features, privacy settings can be restricted to avoid only these strangers. On the other hand, if a friend increases the risk labels of strangers, all of his/her strangers should be avoided.

Privacy risks that are associated by friends' actions in information disclosure has been studied in [14], but the authors work with direct actions (e.g., re-sharing user's photos) of friends, rather than their friendship patterns. Recent privacy research mainly focused on automatically finding the best privacy settings (see [5] for a review) . However, research works have been mainly limited to finding the best privacy settings by observing the interaction intensity of user-friend pairs [4] or by asking the user to choose privacy settings [6]. In contrast, without explicit user involvement, Leskovec et al. [10] have shown that the attitude of a user towards another can be estimated from evidence provided by their relationships with other members of the social network. Similar works try to find friendship levels of two social network users (see [1] for a survey). Although these work can explain relations between social network users, they cannot show how existence of mutual friends can change these relations.

A more detailed and extensive version of this work can be found at http://arxiv.org/abs/1210.3234. This paper is organized as follows. Section II gives an overview of our model, whereas Section III deals with the dataset. In Section IV we discuss the role of profile features in risk labels, whereas Section V shows how impacts of friends are modeled. Section VI gives experimental results, whereas we conclude with Section VII.

II. Overall Approach

Starting from the risk model presented in [3], the goal of this paper is to understand whether friends have negative or positive impacts on risk labels of strangers. At this aim, we have to know what risk label the stranger would receive from the user if there were no mutual friends. This corresponds to the case where a user given label depends only on stranger features. Given a user u and a stranger s, we will term this projected label as the *baseline label*, and show it with b_{us}. For

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.57

instance, assume that if there are no mutual friends, a user u considers all male users as very risky, and avoids interacting with them. In this case, the baseline label for a male stranger s is very risky, i.e., $b_{us} = very\ risky$. However, if the same male stranger s has a mutual friend with user u, we assume that the *user given label*, denoted as l_{us}, might not be equal to the baseline label b_{us} (i.e., $l_{us} \neq b_{us}$), because the mutual friend might increase or decrease the risk perception of the user. The difference between the baseline and user given labels will be used to find out friend impacts.

Finding baseline labels and friend impacts require different approaches. In baseline estimates, we use logistic regression on stranger features, whereas for friend impacts we use multiple linear regression [13]. Overall, we divide our work into three phases as follows:

Transformation: Exploit the risk label dataset from [3] in such a way that regression analyses for baseline labels and friend impacts can find results with high confidence. With this step, we increase the number of labels that can be used to estimate baseline labels and friend impacts.

Baseline Estimation: Find baseline labels of strangers by logistic regression analysis of their features.

Learning Friend Impacts: Create a multiple linear regression model to find friends that can change users' opinion about strangers and result in a different stranger label than the one found by baseline estimation.

III. Transforming Data

Scarcity of user given data labels is also a problem in Recommender Systems (RS) [12] where the goal is to predict ratings for items with minimum number of past ratings. In neighborhood based RS [9], ratings of other similar users are exploited to predict ratings for a specific user. Traditionally, the definition of similarity depend on the characteristics of data (e.g., ordinal or categorical data), and it has to be chosen carefully. At this purpose, we use profile data of friends and strangers in defining similar friends and strangers, respectively. This allows us to learn friend impacts of a user u from impacts of similar friends from all other users. To this end, we transform profile data of friends and strangers in such a way that friends and strangers of different users are clustered into global friend and stranger clusters. Then, we learn friend impacts from clusters.

Clustering Friends

This transformation uses the homophily assumption [11] which states that people create friendships with other people who are similar to them along profile features such as gender, education etc. In other words, we assume that all friends of a user u can be used to judge the similarity of a social network user to u. For example, a user from Milan can have a friend from Milan, whereas a user from Rome can have a friend from Rome. Although these two friends have different hometown values (Milan and Rome), we can assume that both friends can be clustered together because their hometown feature values are similar to those of the users. Thus, we assume that different users will have similar clusters of friends, e.g., friends from the same user's hometown. Then, friend impact values will be correlated with their corresponding clusters, e.g., friends from hometowns will have similar impact values. By considering these, we transform categorical friend values to numerical values in such a way that similarities between friend and user values become more accurate.

More precisely, the transformation of friends' data maps a categorical feature value of a friend, such as hometown:Milano, to a numerical value which is equal to the frequency of the feature value among profiles of all friends of a user. For example, if a friend f has profile feature value hometown:Milano, and there are 15 out of 100 friends with similar hometown:Milano values, hometown feature of f will be represented with $15/100 = 0.15$. After applying this numerical transformation to all friends of all users, we compute a Social Frequency Matrix for Friends (SFMF) where each row represents numerical transformation of feature vector of a user's friend.

***Definition** 1 (Social Freq. Matrix for Friends):* The Social Frequency Matrix associated with a social network $\mathcal{G}$ is defined as $|N| \times |F| \times n$, where N is the set of users in $\mathcal{G}$, $F \subset N$ is the set of users in $\mathcal{G}$ that are friends of at least one user $u \in N$, and n is the number of features of user profiles. Each element value of the matrix is given by:

$$SFMF[u, f, v] = \frac{Sup(\vec{f_v})}{|F_u|}$$

where $F_u \subset F$ is the set of friends of u, $Sup(\vec{f_v}) = \left|\{g \in F_u | \vec{g}_v = \vec{f_v}\}\right|$ and $f \in F_u$, whereas $\vec{g}_v$ and $\vec{f_v}$ show the value of profile feature v for users g and f, respectively.

Having transformed friend data into numerical form, we can now use a clustering algorithm to create clusters of friends. After applying a clustering algorithm to the Social Frequency Matrix for friends, output friend clusters will be denoted by FC.

Clustering Strangers

By clustering friends, we can learn impacts of friends from different clusters, but this raises another question: do friends have impact on all strangers of users? Our assumption is that correlation between stranger and friend profile features can reduce or increase friend impact. With this assumption, we transform strangers' profile data to numerical data and cluster the resulting matrix just like we clustered friends. This clustered stranger representation helps us detect clusters of strangers for whom certain clusters of friends can change risk perception of users the most. We prepare a Social Frequency Matrix for Strangers similar to the one given for friends in Definition 1. Formally, we have $SFMS[u, s, v] = \frac{Sup(\vec{s}_v)}{|F_u|}$, where s is a stranger and $\vec{s}_v$ is the value of profile feature v for s.

Note that we still use values from friend profiles in the denominator to transform stranger data. We use friend profiles because we expect them to be similar to profiles of their own friends (strangers).

We again use the Social Frequency Matrix for Strangers to create clusters of strangers. We will denote these stranger clusters by SC. In our experiments, we used the *k-means* and hierarchical algorithms [7] to produce clusters of friends and strangers.

IV. Baseline Estimation

Baseline estimation analyzes how feature values on stranger profiles bring users to assign specific risk labels to strangers. The baseline estimation process results in baseline labels for each stranger of a given user. These labels are found by using statistical regression methods on already given user labels and stranger profile features. In this section we will discuss this process.

Baseline estimation corresponds to the case where a user would assign a risk label to a stranger without knowing which one of his/her friends are also friends with the stranger. In baseline estimation we use the labels of strangers who have the least number of mutual friends with users. These are the subset of labels which were given to strangers who have only one friend in common with users. In what follows, we will use *first group dataset* to refer to these strangers.

In our approach, we use logistic regression to learn the baseline labels from available data. This allows us to work with categorical response variables (i.e., one of the tree risk labels). Stranger features are used as explanatory (independent) variables and risk labels as the response (dependent) variable which is determined by values of explanatory variables (i.e., feature values).

For example, for a specific stranger, logistic regression can tell us that risk label probabilities of the stranger is distributed as %0.9 very risky, %0.09 risky and %0.01 not risky. As we can compute baseline label in real values, a stranger $s \in S$ is assigned a baseline label by weight averaging the probabilities of risk labels.

V. Friend Impact

In computing friend impacts, we use multiple linear regression [13], which learns friend impacts by comparing baseline and user given labels to strangers. To this end, we define an estimated label parameter to use in linear regression as follows:

***Definition** 2 (Estimated label):* For a stranger s and a user u, an estimated label is defined as:

$$\hat{l}_{us} = b_{us} + \sum_{FC_i \in FC} FI(FC_i, SC_j) \times Past(u, s)$$

where $\hat{l}_{us}$ and b_{us} are estimated and baseline labels for a stranger s, and s belongs to the stranger cluster $SC_j \in SC$. $Past(u, s)$ denotes an intermediary value based on stranger labels given by user u, whereas $FI(FC_i, SC_j)$ represents impact of a friend f from a friend cluster FC_i on the label of stranger s from a stranger cluster SC_j.

In the rest of this section, we will define the $Past(.,.)$ and $FI(.,.)$ parameters, and explain how they are used to compute friend impacts.

A. The Past Labeling Parameter

The need for this parameter arises from the fact that baseline estimation is computed from labels of all strangers who have only one mutual friend with user u (i.e., first group dataset), and it tends to be a rough average. To overcome this, a subset of strangers, who are very *similar* to s and who have been labeled in the past by u, are observed and the baseline label is increased or decreased to make it more similar to the user given labels of these strangers. Here, we consider two factors: how many similar strangers should be considered in this adjustment and what is an accurate metric for finding similarity of two strangers? For the first question, we use the computed stranger clusters. For a stranger s, similar strangers from the first group dataset are those (i) that are labeled by the same user u, and (ii) that belong to the same stranger cluster with s. For the second question, we use the profile similarity measure by Akcora et al. [2]. This measure assigns a similarity value of 1 to strangers with identical profiles, and for non-identical profiles the similarity value is higher for strangers whose profile feature values are more common in profile features of u's friends. Formally, we define the past labeling as follows:

***Definition** 3 (Past Labeling Parameter):* For a given user u and stranger s, the past labeling parameter is defined as:

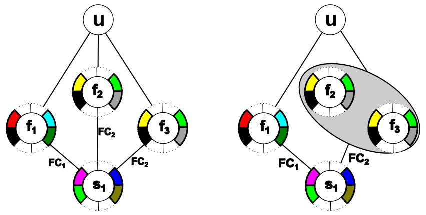
$$Past(u, s) = \frac{1}{|SC_i|} \sum_{x \in SC_i} PS(s, x) \times (l_{ux} - b_{ux})$$

where strangers s and x belong to the same stranger cluster SC_i, $PS()$ denotes the profile similarity between them, l_{ux} is the user given label of stranger x, and b_{ux} is the baseline label of x.

B. The Friend Impact Parameter

$FI(f, s)$ is used to show impacts of mutual friends on the risk label given to s by u. In modeling friend impacts, we wanted to see how friends from different clusters changed the baseline label. If there is at least one mutual friend from a friend cluster FC_i, we say that friend cluster FC_i may have impacted the label given to the stranger s. For the cases where a stranger s has two or more mutual friends from a friend cluster FC_i, we experimented with both options for $FI(f, s)$. Next, we will explain these options.

1) Multiple Impact for the Friend Cluster: In our first approach, we assume that a bigger number of mutual friends from friend cluster $FC_i \in FC$ will impact user labeling. This approach is shown in Figure 1(a). Assume that from a friend cluster $FC_i \in FC$, we are given the set of mutual friends $MF_i = \{\forall f | f \in FC_i, f \in \{F_u \cap F_s\}\}$ of user u and stranger s. We define the impact of friend cluster FC_i on the label of stranger $s \in SC_j$ as follows: $FI_2(FC_i, SC_j) = |MF_i| \times I_{FC_i,SC_j}$ where I_{FC_i,SC_j} is the impact of a cluster $FC_i | f \in \{FC_i \cap MF_i\}$ on the label of stranger $s \in SC_j$. Note that this impact (I_{FC_i,SC_j}) is the unknown value that our system will learn.



(a) Multiple impacts for a friend cluster.

(b) Single impact for a friend cluster.

Fig. 1. Friend impact definitions by considering the number of friends from the same cluster. In the single impact definition, two friends do not increase the friend impact.

2) Single Impact for the Friend Cluster: In the second approach, we assume that a bigger number of friends from the same cluster does not make a difference in user labeling; at least one friend from the cluster is required, but more friends do not bring additional impact. This approach is shown in Figure 1(b), where friends are shown with their cluster ids, and two friends from friend cluster FC_2 bring a single impact. Assume that from a friend cluster $FC_i \in FC$, we are given a set of mutual friends of user u and stranger s. We give the impact of friend cluster FC_i on the label of stranger s as follows: $FI_1(FC_i, SC_j) = I_{FC_i,SC_j}$ where I_{FC_i,SC_j} is the impact of a friend cluster FC_i on label of stranger $s \in SC_j$.

These different friend impact approaches change the model by including different numbers of friend impacts. The unknown impact variable I_{FC_*,SC_*} is learned by the least squares method [8].[1]

In the experimental results, we will discuss the definition that yielded the best results.

VI. Experimental Results

In this section we will validate our model assumptions, and then continue to give detailed analysis of performance under different parameter/setting scenarios.

A. Validating Model Assumptions

Before finding friend impacts, we validated our model assumption (i.e., mutual friends have an impact on the risk label of a stranger) by using logistic regression on the whole dataset (4013 stranger labels and profiles). For this, we included *the number of mutual friends* as a parameter, and computed the *significance*[2] of model parameters. In overall regression, photo visibility, wall visibility, education and work parameters were excluded from the model because they were found to be non-significant. For significant parameters, $Pr(> |t|)$ values are shown in Table I.

In the regression, there are two friend related parameters: the number of mutual friends and the friendlist visibility. Differing from the number of mutual friends, friendlist visibility is a categorical variable with 0 value when the stranger hides his/her friendlist from the user and 1 otherwise. From Table I,[3] we see that seeing a stranger's friendlist increases the probability of the stranger getting label 1, whereas it is not an important parameter for label 3. Our main focus in regression analysis was to verify that the number of mutual friends parameter is significant. We found that an increasing number of mutual friends indeed helps a stranger get label 1, and decreases the probability of getting label 3. This result tells us that friends have an impact on user decisions and our assumption about the existence of friend impacts holds true. After validating our model assumption, we continue to the baseline label estimations.

TABLE I
Regression results for all data points.
p-value=2.22e-16. Total N=4013.

	Label 1	Label 3
Intercept	-1.2668850***	-0.7626810***
	(0.146)	(0.138)
Mutual friends	0.0379547***	-0.0467834***
	(0.008)	(0.012)
Gender	-0.3696749**	0.3480055**
	(0.118)	(0.113)
Friendlist visibility	0.6203365***	-0.0642952
	(0.125)	(0.118)
Locale	0.6167273***	0.7070663***
	(0.180)	(0.172)
Location	l0.1347104	0.2708697*
	(0.128)	(0.125)
N	1116	1161

TABLE II
Regression results for the first group data points.
p-value = 5.6701e-11. Total N=1520.

	Label 1	Label 3
Intercept	-2.5400***	-0.8661***
	(0.6305)	(0.2791)
Gender	-1.1026**	0.6985*
	(0.4108)	(0.3350)
Friendlist visibility	0.4705*	0.5214
	(0.2075)	(0.1706)
Wall	0.4173•	-0.1595
	(0.2463)	(0.2262)
Photo	1.9425**	0.1361
	(0.6093)	(0.2339)
Locale	0.1446	0.5846*
	(0.2881)	(0.2277)
N	278	588

B. Training for Baseline

Baseline calculation predicts labels for strangers without friend impacts. For this purpose we take strangers who have one mutual friend with users ($|MF| = 1$) into a new dataset (first group dataset), and train a logistic regression model. Logistic regression on the first group dataset finds how stranger features bring users to label strangers. Table II shows model parameters and their corresponding p-values.

In Table II, we see that when users label the first group strangers, photo and wall visibility are significant parameters. If these items are visible on stranger profiles, the probability

[1] The least squares method provides an approximate solution when there are more equations than unknown variables.

[2] Significance is measured by p-values. The p-value is the probability of having a result at least as extreme as the one that was actually observed in the sample. Traditionally, a $p - value$ of less than 0.05 is considered significant.

[3] Reference category for the equation is label 2. Standard errors in parentheses. Significance codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

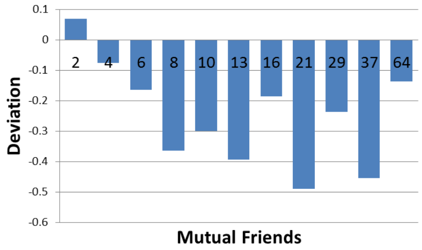


Fig. 2. Deviation of user given labels from baseline labels. Values in the x-axis are the number of mutual friends between a stranger and user.

of strangers getting label 1 increases. In the whole dataset (see Table I), these two parameters were found to be insignificant.

After computing a baseline label for all strangers, we use the difference between user given and baseline labels ($l_{us} - b_{us}$) to model the friend impact. These differences (deviations from the baseline label) are shown in Figure 2. In the figure, we see that user given labels are lower than the computed baseline label, which shows that in overall friends have positive impacts (i.e., thanks to mutual friends, users assign lower risk labels to strangers.). Overall, we found that there does not exist a linear relation between the number of mutual friends and the deviation values. This non-linearity changes how we define the impacts of friend clusters. In Section V we gave two definitions for friend impacts (see Figure 1) to account for deviations from the baseline label.

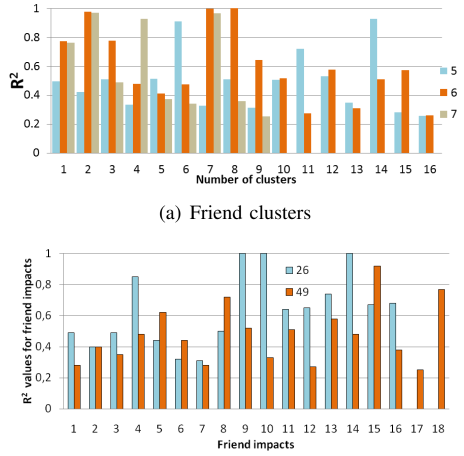
In multiple friend impacts we assumed that more mutual friends from a friend cluster bring additional impacts. On the other hand, in single friend impact one friend was enough to have the impact of a friend cluster. This finding implies that more friends of the same cluster do not provide any benefits to strangers on Facebook and mutual friends from different clusters are more suitable to change the user's risk perception about a stranger. We believe that this can be generalized to other undirected social networks.

In the rest of the experiments, we will give the results computed by using the single friend impact definition. We will now explain the model performance under different clustering settings.

C. Clustering

For clustering 12659 friends, and 4013 strangers we experimented with k-means and hierarchical clustering algorithms. In our experiments with different numbers of final clusters, the k-means algorithm yielded the best results for friend clustering, whereas hierarchical clustering was better for stranger clustering. Due to space limitations, we will omit hierarchical clustering results for friends and k-means results for strangers.

Friend Clustering: In Figure 3(a) we show the adjusted coefficient of determination (R^2) of our multiple regression model with different k values for friend clustering. The x-axis gives the number of stranger clusters for which at least one friend cluster has an impact.



(a) Friend clusters

(b) Stranger clusters

Fig. 3. Coefficient of determination (R^2) values for friend and stranger clusters.

For two k values, 5 and 6, we have the best results. Our model hence suggests that friends of social network users can be put into 5 or 6 clusters when considering how much they can affect user decisions on stranger labeling.

Stranger Clustering: In Figure 3(b) we see that 26 stranger clusters lead to R^2 values close to 1, and we can find friend impacts in 16 out of 26 stranger clusters.

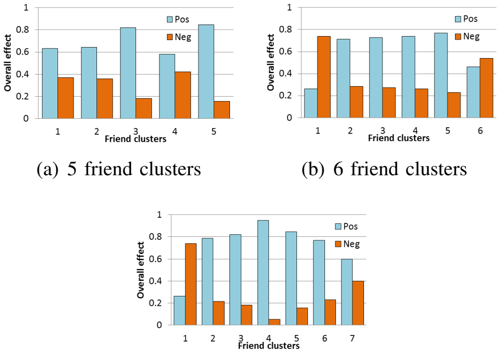
Cross Validation: After clustering and prior to learning friend cluster impacts, we prepare a test set for validating our model. We remove 10% of strangers from stranger clusters and set those aside as the test strangers (T). Once friend impacts are found for stranger clusters, we plug in the set of test strangers, and calculate the root mean square value (RMSE) of their labels.

Cross validation results for different numbers of stranger clusters is detailed in Table III by using 6 friend clusters. The first row of the table shows the number of stranger clusters, whereas the second row shows the average R^2 values in these clusters. In the third row, we show the median size of stranger clusters; with increasing numbers of clusters, the number of strangers in each cluster decreases. In the case of 158, the average number of strangers in a cluster is reduced to 7, and this results in a poor performance because the model cannot have enough data to learn friend impacts on stranger clusters. The average number of validation points are shown in the fourth row. An increasing number of stranger clusters results in fewer validation points because some clusters have less than 10 strangers themselves. In the fifth row, the root mean square values are shown for these validation points. In 26 stranger cluster our model yields the best R^2 and $RMSE$ pair results.

These experimental results suggest that the optimal number of stranger clusters (26) is bigger than the optimal number of friend clusters ($k = 5, 6$). We explain this by the fact that although users can choose friends of specific characteristics, they cannot do so with strangers. As a result, strangers are

TABLE III
PERFORMANCE VALUES FOR DIFFERENT NUMBERS OF STRANGER CLUSTERS.

Cluster count	8	26	49	82	158
R^2	0.51	0.64	0.48	0.54	0.45
Median Size	62	25	16	12	7
Validation points	179	99	69	48	27
RMSE	0.35	0.45	0.62	0.97	0.94



(a) 5 friend clusters

(b) 6 friend clusters

(c) 7 friend clusters

Fig. 4. Percentage of positive and negative impact values for friend clusters.

more diverse than friends, and they need to be clustered differently from friends.

D. Friend Impacts and Risk Labels

In this section we will give computed friend cluster impacts, and show how friends are assigned risk labels.

The rationale behind clustering was to observe different friend cluster impacts on different stranger clusters. Although a friend cluster can have an overall positive impact (i.e., reduces the risk label of most strangers), friend clusters might have different signs and multitudes of impact values on stranger clusters. In Figure 4 we show how different friend clusters can have positive and negative impact values for different k values (number of friend clusters). As seen in Figure 4(a), when we increase the number of friend clusters from $k = 5$ to $k = 6$, positive and negative impact frequencies change for each cluster because either friend clusters became more homogeneous or some clusters did not have enough data points to learn from. Figure 4(b) shows two friend clusters with overall negative impacts (friend clusters 1 and 6). Figure 4(c) shows the positive and negative impact frequencies for $k = 7$, where frequencies are more emphasized for negative and positive impacts of a cluster. Note that the number of overall negative clusters is reduced from 2 to 1 here. Similar to a transition from 5 to 6 clusters, friends of two negative clusters might be put into the same cluster (cluster 1) or there were no longer enough strangers for some friend clusters to learn a negative impact.

The existence of both positive and negative impact values for each friend cluster confirms our intuition that impacts of friend clusters vary depending on a stranger cluster. A friend is assigned a higher risk label when a friend cluster has a big percentage of negative impact values.

VII. CONCLUSION AND FUTURE WORK

In this work, we analyzed how the risk labels of friends of friends can be used to compute risk labels of friends. We found that the number of mutual friends is not very important to change the risk perception of a user towards a friend of friend. On the other hand, having different types of mutual friends (i.e., friends from different friend clusters) with a friend of friend plays a bigger role in users' risk perception. In the future, we want to create sets of global privacy settings by using our risk model, so that privacy settings can be automatically applied to different social network users.

REFERENCES

[1] W. Ahmad and A. Riaz. Predicting friendship levels in online social networks. Master's thesis, Blekinge Institute of Technology, 2010.

[2] C. Akcora, B. Carminati, and E. Ferrari. Network and profile based measures for user similarities on social networks. In *Information Reuse and Integration (IRI), 2011 IEEE International Conference on*, pages 292–298. IEEE, 2011.

[3] C. G. Akcora, B. Carminati, and E. Ferrari. Privacy in social networks: How risky is your social graph? In *The 28th IEEE International Conference on Data Engineering*, 2012.

[4] L. Banks and S. Wu. All friends are not created equal: An interaction intensity based approach to privacy in online social networks. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 970–974. IEEE, 2009.

[5] M. Beye, A. Jeckmans, Z. Erkin, P. Hartel, R. Lagendijk, and Q. Tang. Literature overview - privacy in online social networks, October 2010.

[6] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360. ACM, 2010.

[7] G. Gan, C. Ma, and J. Wu. *Data clustering*. SIAM, Society for Industrial and Applied Mathematics, 2007.

[8] B. Jiang. On the least-squares method. *Computer methods in applied mechanics and engineering*, 152(1-2):239–257, 1998.

[9] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1, 2010.

[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 641–650, New York, NY, USA, 2010. ACM.

[11] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 1:415–444, 2001.

[12] P. Melville and V. Sindhwani. Recommender systems. *Encyclopedia of Machine Learning*, 1:829–838, 2010.

[13] R. Myers. *Classical and modern regression with applications*, volume 488. Duxbury Press Belmont, California, 1990.

[14] K. Thomas, C. Grier, and D. Nicol. unfriendly: Multi-party privacy risks in social networks. In *Privacy Enhancing Technologies*, pages 236–252. Springer, 2010.