A Machine Learning Based Approach for Predicting Undisclosed Attributes in Social Networks

Gergely Kótyuk
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
Email: kotyukg@crysys.hu

Levente Buttyán
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
Email: buttyan@crysys.hu

***Abstract*—Online Social Networks have gained increased popularity in recent years. However, besides their many advantages, they also represent privacy risks for the users. In order to control access to their private information, users of OSNs are typically allowed to set the visibility of their profile attributes, but this may not be sufficient, because visible attributes, friendship relationships, and group memberships can be used to infer private information. In this paper, we propose a fully automated approach based on machine learning for inferring undisclosed attributes of OSN users. Our method can be used for both classification and regression tasks, and it makes large scale privacy attacks feasible. We also provide experimental results showing that our method achieves good performance in practice.**

I. Introduction

The notion of social network was introduced in 1954 by J.A. Barnes, sociologist. Since then a new form of social networks has emerged: online social networks (OSN), e.g. Facebook, LinkedIn, etc. OSNs provide a great amount of data for social network researchers, while at the same time, they raise privacy questions, making it an interesting research topic for the computer security community too.

Insurance companies might want to find out the age of OSN users, the government might be interested in their political affiliation, or spammers might be eager to any user attribute in order to be able to send targeted advertisements. Against these threats OSNs provide privacy protecting mechanisms, which typically let users control the visibility of their attributes.

Although users can hide the attribute values they want to keep private, visible attributes, friendships, group memberships, etc. do carry information about those private attributes, and machine learning methods provide efficient means to infer hidden values from information available in the social network.

In machine learning there are two kinds of tasks depending on the type of the target variable. The task is classification if the target variable is nominal (e.g. marital status), and regression if it is numerical (e.g. age). Previous works have focused largely on classification. In this paper we present an attribute inference attack based on the concept of Multi Layer Perceptron (MLP). Our method is capable of both classification and regression. To our best knowledge, this is the first work in the field that deals with regression.

For inferring a private attribute value several sources of information are available. Some of them are useful, while others are useless. For example, when classifying the marital status, age might carry useful information, but postal code probably does not. If there are several possible input variables, selecting the right ones is a complicated task for humans, thus, should be automated. We present a correlation matrix based approach for automatically selecting useful input variables.

In our model inference of a private attribute of a user is based on his public attributes and his friends' public attributes. Since users might have a great number of friends, and their number varies from user to user, it is not possible to have as many MLP inputs as the attributes of all of the users friends. To solve this problem, we propose a way for aggregating friends' attributes in some (a fixed, and not too large number of) variables.

In summary, our main contributions are the following:

- we propose an MLP-based system for inferring hidden attributes in online social networks,
- our method can handle both classification and regression task,
- we propose a way of aggregating friends' attributes,
- we present a correlation-matrix-based method for fully automated selection of useful input variables of the neural network,
- the input variables of the MLP are chosen from a large set: when inferring a private attribute of a given user, any of the user's public attributes, and any of the aggregates of its friends' attributes can be selected.

The rest of this paper is organized as follows. In section II, we present the related work. We describe our inference system in general in section III, and discuss the way it was implemented in our case in section IV. In section V, we provide experimental results, and in section VI, we conclude the paper.

II. Related Work

Domingos et al. [1] applied social network analysis for marketing purposes. They considered customers as nodes in a social network and modeled their influence on each other as a Markov random field. Using the Markov random field, they calculated the probability of the event that a user buys a product, given that some marketing actions are taken (e.g. discount is offered). They tested their method on EachMovie, a collaborative filtering database for marketing motion pictures.

He et al. [2] mapped social networks to Bayesian networks. To infer the value of a given attribute A of a given user U a Bayesian network is constructed: each node corresponds to a user and represents its attribute A, while the links represent friendships among users. They considered multiple hop inference too.

Johnson et al. [3] made the assumption that people are like their friends, therefore a hidden attribute of a user can be inferred from attribute values seen at his friends. They managed to manually predict quite accurately whether a Facebook user was homosexual by looking at friends' gender and sexuality. The experiment highlighted the fact that friendships carry a large amount of information about users.

Lindamood et al. [4] present a modification of Naïve Bayes classification which uses both node attributes and link structure. To protect privacy they propose the removal of the K most representative attributes from the graph, globally, and the L most representative links from each node, locally.

Mo et al. [5] proposed using Semi-Supervised Learning (SSL), which is a machine learning framework derived by combining supervised learning with unsupervised learning. It uses a small set of labeled data and a much larger set of unlabeled data. Thus it suits well social network data, which usually contain little publicly available information and much hidden information. They provide two specific attack models: (1) a graph-based attack model, based on local and global consistency, and (2) a co-training model consisting of a graph based SSL algorithm for learning link structure information, and a supervised learning algorithm for personal information found in attributes.

Mo et al. [6] presented another attack using graph-based SSL. Besides the requirements of local and global consistency as presented in [5], the requirement of community consistency was added. Communities, created with clustering algorithms, are groups of users who have strong connection with each other. They proposed a closed-form and an iterative learning algorithm, which take local, global and community consistency into account.

Social network graphs can be clustered to communities, i.e. groups of users who are more tightly connected than the surrounding graph. Users in the same community often share the same attribute values, on which user attribute inference methods can be built. Mislove et al. [7] inferred attributes based on a known global community detection algorithm with the minor modification that they used *normalized mutual information* metric for measuring the similarity of community structures. Their main contribution is the attribute inference based on a new local community detection algorithm, which uses the metric *normalized conductance*. They used two Facebook datasets for testing and succeeded to infer user attributes with high accuracy in certain cases, depending on the strength of the community.

Thomas et al. [8] examined how the lack of *joint* privacy control reveal sensitive information. They aggregate information disclosed in this manner and infer attributes: gender, political views, religious views, relationship status (single label classification), favorite music, movies, television shows, books (multi label classification). They propose two classifiers. One of them is based on friendship information, it uses multinomial logistic regression for single label tasks and linear regression for multi label classification tasks. The other classifier is based on the Facebook wall content, in which word frequencies are counted and used as input of a multinomial logistic regression. They showed how Facebook's privacy model can be adapted to enforce multi-party privacy.

Social network users can join to groups (e.g. group for a certain music band, fans of a certain actor, etc.), which indicate their interests. The main contribution of Zheleva et al. [9] is making use of such group information. Groups are assumed to be homogeneous, thus members of the same group should have similar attribute values, which can be used to infer hidden attributes of users. They proposed several methods, the best of which proved to be the one based on Support Vector Machine (SVM). There are several differences between their work and ours. At first, our method is capable of both classification and regression, while they considered classification only. Another major difference is that in our method potentially each user attribute is used as neural network input, while they used only the inferred attribute values observed at other users (at friends, in groups). Our solution includes a correlation-matrix-based method, which allows automatic variable selection. Their main contribution is highlighting the importance of group information, while ours is presenting a fully automated inference system, which allows both classification and regression. Test results show that they inferred gender in Facebook with similar accuracy as we did in iwiw, a Hungarian OSN (see section V).

III. Approach

A. Overview

A social network can be represented as a directed graph, where the nodes represent users, and the edges represent friendships among them. User attributes can be modeled mathematically as node labels. Edges can also be labeled if friendships has attributes (e.g. lifetime of friendship, type of friendship, etc.).

Our goal is to give a general approach to predict any undisclosed attribute values on user profiles in an automated way. Formulated in an other way, our goal is to fill in empty fields in the entire social network based on the information available in the network.

We propose a Multiple Layer Perceptron (MLP) based inference system, which is capable of inferring attributes without any user interaction. Figure III-A gives an overview of the system.

In order to be able to select the input variables for the inference of output variables a correlation matrix is calculated at the first step, which describes the strength of relationship between each pair of variables. The remaining steps are executed for each inferred attribute. In the second step, the proper input variables for the current output variable are selected based on the correlation matrix. In the third step, a neural network (MLP) is created. Note that different neural networks must be

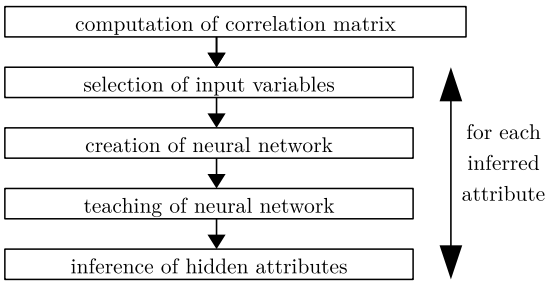


Fig. 1. System overview

created for inferring different attributes. In the fourth step,- the MLP is trained, and finally the hidden values are inferred. For the training we used error backpropagation, the most common training method for MLPs. For the last two steps the whole dataset (the data of all of the users observed) is used. Then the process is repeated from the second step for the next attribute to infer.

B. Types of variables and their usability

From a machine learning point of view, variables (user attributes) can be sorted to the following three groups based on their values:

- nominal with few possible values
- nominal with many possible values
- numerical

Input as well as output variables of different groups must be handled in different ways when constructing an MLP.

If the output variable is numerical (e.g. age), the task is called regression (alias function approximation) regardless of the type of input variables. This time the output layer of the MLP consists of one single neuron, and the output is a continuous number in (-1, 1) or (0, 1), which must be scaled and/or quantized if needed.

If the output variable is nominal (e.g. marital status), the task is classification, regardless of the type of input variables. In this case, if the target variable can take n different values, the output layer consists of n neurons, each one corresponding to a possible value of the output variable. The (continuous) output of the i^{th} neuron of the output layer represents the extent to which the input seems to belong to the i^{th} category. The maximum is selected among the outputs of the neurons, and the corresponding nominal value is returned as nominal output. This works well for nominal variables with few possible values, but the approach becomes infeasible when the number of possible values is large, because it would require too many output neurons in the output layer, which would result in a large and slow MLP. Thus nominal output attributes with many possible values cannot be handled with MLPs.

The situation is similar at the input of the MLP. If the input variable is numeric, it can be used as an input of the MLP without any modification. For a nominal input variable, a dummy variable is created for each value of the nominal variable. These dummy variables are then used as MLP input variables.

For the same reasons as for output variables, nominal variables can only be used as input if the number of possible input values is small.

C. The aggregation of friends' attributes

As explained above, a set of attributes is inferred one by one. For each inferred attribute, an MLP is created, which is then trained and tested using the dataset of all users[1]. Since the number of friends changes from user to user (and this number is large), while the MLP has a fix (and not too large) number of inputs, attributes of friends cannot be directly used as MLP inputs. Hence the attributes of all of the friends of a user must be aggregated to a fix number of variables. We propose different methods for aggregation depending on the type of attribute variables.

If the attribute is nominal with few possible values (categories), then for each category, the number of friends belonging to that category can be counted. This category-by-category enumeration leads to (not too many) variables, which can be used as MLP input. For example, marital status might have five different values: single, in a relationships, married, divorced, widow. To aggregate the marital status of the friends of a user, the number of friends who are single, who are in a relationship, etc. is counted. This way five variables are created, which describe the marital status of the friends of the current user without any information loss. These are variables, since their values vary from user to user. The five variables can be used as input variables of an MLP.

If the attribute is nominal with many possible values, then the above presented method of aggregation is not possible. In this paper do not handle this case, and leave the problem for future work.

If the attribute takes its value from an ordered set, then the distribution of the values observed at the friends of a given user can be calculated. However the distribution still cannot be used as MLP input directly, thus the information carried by the distribution should be handed over the MLP through a number descriptors of the distribution. We used the following statistical measures to describe distributions [10]: mean, deviation, variance, skewness, kurtosis, entropy, energy. These measures are calculated over the friends of a user. Since the values of the measure vary from user to user, each measure is a variable. This way an attribute which takes its value from an ordered set can be aggregated over friends in seven variables (mean, deviation, etc.), which can be used as MLP input.

D. The selection of input variables

Since social networks might provide users with many attribute fields and aggregation of friends' attributes introduces several new attributes, the set of possible input variables for a given target variable is typically quite large. However most of the possible input variables do not carry any information about the output variable. The presence of such input variables

[1]this is actually not necessary, a sufficiently large set would do

makes the learning and testing process slow, and brings noise to the input, thus they should be eliminated.

The large number of possible input variables makes it infeasible to manually select a right subset of input variables, hence an automated method is needed for that task.

We propose computing correlation between each pair of attribute variables, and for a given target variable, let the input variables be the ones which have their correlation coefficient with the target variable above a threshold value.

To understand the way of computing the correlation coefficients, imagine a dataset with observation of n attributes of N users. This can be stored in a table of n columns (attributes) and N rows (users). The correlation coefficient between column i and j is calculated, and written to cell (i, j) of the correlation matrix. This way the correlation matrix consists of the correlations among attribute variables. The correlation ρ between variables X and Y are calculated as $\rho = \frac{\sigma(X,Y)}{\sqrt{\sigma^2(X)\sigma^2(Y)}}$, where $\sigma(X, Y)$ denotes the covariance between variables X and Y, and $\sigma^2(X)$ denotes the variance of X. The covariance $\sigma(X, Y)$ is calculated as $\sigma(X, Y) = \frac{1}{n-1}\Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$, where n is the number of elements in the dataset and $\bar{x}$ is the average of $x_1, x_2, ..., x_n$, while variance is $\sigma^2(X) = \sigma(X, X)$.

IV. Implementation

We tested our system with a dataset obtained from iwiw[2], the largest Hungarian online social network with more than 4 million registered users. In iwiw, access to user profiles is not limited (i.e. all of the attributes are public), and it does not control the download rate, therefore, it was relatively easy to obtain a pretty high quality dataset. To gather training points (user data) we wrote a crawler, which was distributed over several computers in order to increase performance.

Using the crawler we downloaded the datasheets (attributes) and friendships of a large amount of users. However for MLP training we used the data (attributes and friendships) of core users only, which are defined as users, all of whose friends' datasheets were downloaded. To reach the maximum number of core users, the crawler downloaded user data in a breadth-first search order.

The distributed downloading system was implemented in Python, because it provides convenient methods for accessing websites and processing text. We downloaded altogether 923 914 user profiles with a core size of 13 322 profiles. We used only the core users for MLP training.

We implemented the inference system in Matlab. The dataset could be read from the database management system with a Java connector. Each row of the dataset consists of the attributes of a user. Therefore the columns of the dataset correspond to attribute variables.

MatLab's *corrcoef* function calculates correlation coefficients among variables. This function, provided with the dataset, calculated the correlation matrix, and the corresponding probability matrix, with which the hypothesis of non-existing correlation can be tested. We always used significance level 95% when deciding the hypothesis. The input variables for the given target variable were selected using the correlation matrix, restricted with the results of hypothesis testing.

The Neural Network Toolbox of Matlab provides an off-the-shelf implementation of MLP. It creates the neural network with the desired number of neurons, and provides several training algorithms too. We used the Levenberg-Marquardt training algorithm, which is based on the most common error back propagation method. In regression tasks, we used the mean absolute error (mae) as performance function instead of the default mean squared error (mse), because mean absolute error can be directly interpreted. The number of neurons in the hidden layer was 5 in all of our test cases.

The dataset was randomly divided to the three subsets introduced in the previous section as follows:

- training set: 60%
- validation set: 20%
- test set: 20%

During the testing of the neural network (using the test set), the network output was compared to the target values of the attributes, which were available in the OSN itself, and downloaded by us.

V. Experimental Results

We inferred the following attributes: age, gender, marital status. Age inference is a regression task, while gender and marital status inference are classification tasks.

We compare the test results of our prediction system with the performance of naïve prediction algorithms, or to results of other works when it is possible. In the case of classification, the naïve algorithm decides either for the most frequent value among the friends or randomly, depending on which method is better. In the case of regression, the naïve algorithm gives the average of the attribute values observed at friends as prediction. For example if the average age of a user's friends is 30 years, then the naïve algorithm predicts 30 for the age of this user.

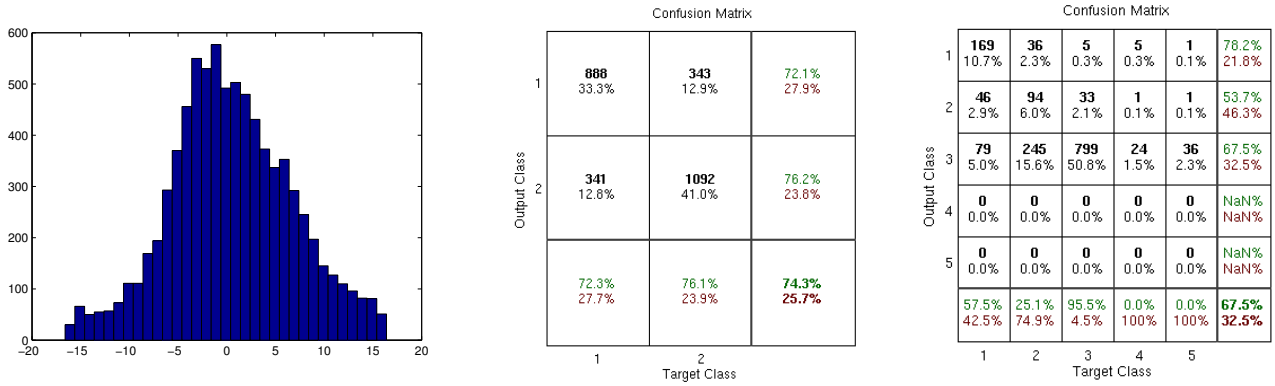
In the following, we describe how we inferred "age", "gender" and "marital status".

A. Estimating the Age

The variables most strongly correlating with age were user's marital status (single or married values) and the average age of friends, as shown in table I. The sign of "marital status single" is negative, which means if the user is single, then its age is usually less. Since the sign of "marital status married" is positive, the married status of the user means that the user is older. The friends' average age is in the strongest correlation with age. It is also a positive correlation, thus the older a users' friends are, the older the user should be.

Figure 2a shows the histogram of age prediction errors in the test case, where lowest and highest 2.5% of the error distribution are not shown. In the case of most of the users the error of the regression is very small (close to zero). There

[2]www.iwiw.hu



(a) Histogram of age prediction error without outliers (b) Confusion matrix of gender (c) Confusion matrix of marital status

Fig. 2. Mean absolute error of age regression and confusion matrices of gender and marital status classification

attribute	correlation
whether marital status is single	-0.52
whether marital status is married	0.46
the average of friends' age	0.7

TABLE I
CORRELATION OF VARIABLES WITH AGE

attribute	correlation
whether the user is single	0.14
number of female friends	-0.2
number of German-speaking friends	-0.09
number of French-speaking friends	-0.09
average of number of languages friends speak	-0.09

TABLE II
CORRELATION OF VARIABLES WITH GENDER

are a few users, whose age was estimated with a large error, which increases the mean absolute error significantly.

The mae of our regression system turned out to be 4.76, while for the naïve regression it is 8.52. It means that if we ask the MLP to infer the age of a given user, then its expected error is 4.76 years, while for the naïve regression it is 8.52 years. As one can see the error of our system is much smaller than that of the naïve regression. Repeating the test 1000 times, each time with different random initial MLP weights, we saw that in 95% of the cases the mae was smaller than 5.15.

Since our work is, to our best knowledge, the first one dealing with regression, our age inference results cannot be compared with results from prior work.

B. Estimating the Gender

The most strongly correlated variables with gender were "whether the user is single", the number of female friends, the number of German-speaking and French-speaking friends, and the average number of languages spoken by the user's friends, as illustrated in table II.

Since inferring gender is a classification task with two classes, the neural network has two neurons at the output layer. The correlation coefficients shown in table II are calculated with the variable for male gender. The correlations with female gender are the same values multiplied by (-1).

Note the correlation coefficient of 0.14 between the variables "the user is male" and "the user is single". This high correlation is surprising, because intuitively the same number of male and female users are expected to be in any kind of relationships, and the number of male and female users is about the same in the dataset (actually there are a bit more female users). The explanation of the positive correlation could be that male users more likely indicate their "single" marital status than female users.

The negative correlation with the number of female friends means that users have more friends of their own gender.

The last three, language-knowledge variables have little correlation with the gender, but we get much worse inference results if we omit them. Supposing the correlation coefficient indicates real connection, the negative sign means that the (female) friends of female users tend to speak more languages than the friends of male users, including German and French.

Figure 2b depicts the confusion matrix of gender classification. The first two rows of the matrix contain the output class (the prediction of the MLP for gender), while the first two columns contain the target class (the correct value of attribute gender). The bottom row is a summary of the rows above. Similarly the right-most column is a summary of the columns to the left. Cells in the main diagonal of the matrix show the number and the ratio of correct classifications. For example the number of cases when the gender was male (class 1) and the MLP predicted male (class 1) correctly was 888. The number of cases when the gender was male (class 1) but the MLP predicted female (class 2) incorrectly was 341. The ratio of correct classifications when the gender was male was 72.3%. The most interesting cell is the bottom right one, which tells that the ratio of incorrect classification was 25.7% in the entire dataset. Performing 1000 tests, each one with different random initial MLP weights, in 95% of the cases the misclassification rate was smaller than 46%.

The naïve predictor, which decides for male if the user has more male friends than female friends and vice versa, classified 29.65% of the dataset incorrectly. Our method performed a bit better with 25.7% misclassification.

Zheleva et al. [9] also predicted gender, and their best result was a misclassification of 27.5% on the whole dataset and 22.8% on the homogeneous half of the dataset. Thomas et al. [8] inferred gender too. Their best ratio was 23.71%

attribute	correlation
age	0.52
number of married friends	0.32
average of friends' age	0.43

TABLE III
CORRELATION OF VARIABLES WITH MARITAL STATUS

misclassification. The main difference of our work and [9] is that they leverage group information, while we do not, and unlike them we do use the observed values of potentially every attribute. In [8] a different approach is used: the authors leverage on information obtained from scattered references, like being mentioned in a story, tagged in a photo, etc.

C. Estimating the Marital Status

The attribute "marital status" shows the strongest correlation with age, number of married friends, and the average age of friends, as shown in table III.

Marital status might have five values: "single", "in relationship", "married", "divorced", "widow", thus five binary valued variables were created, for which five neurons were placed at the output layer. For selecting the input variables, we analyzed the correlation with any of the five created target variables. Table III shows the attribute variables that have the highest correlation with the marital status. The numerical values show the corresponding maximum correlation coefficients over the five created target variables.

Figure 2c depicts the confusion matrix of the MLP classification. Its fields have the same meaning as in the case of gender with the difference that now there are five created target variables, thus the number of rows and columns of the matrix is larger. The bottom-right cell tells that 32.5% of the dataset was classified incorrectly. In 95% of the 1000 tests performed, the misclassification rate was smaller than 50%. The naïve random algorithm, classifies 80% of the dataset incorrectly. Our method clearly outperforms this naïve algorithm.

D. Summary

The results of the proposed inference system compared to the naïve methods is summarized in table IV. In age regression and marital status classification we managed to achieve much better results than the naïve algorithm. In gender classification our results were similar to that of Zheleva et al. [9] and Thomas et al. [8], and a bit better than the naïve algorithm. The main advantage of our method is using potentially each attribute as input, and selecting automatically the ones that correlate the most with the target attribute. This can overcome the lack of group information, which was leveraged in [9], and the ignorance of using various information sources, such as being mentioned in a story, tagged in a photo, etc. [8].

attribute	error of MLP (mae / misclassification rate)	error of naïve predictor (mae / misclassification rate)
age	4.76	8.52
gender	25.7%	29.65%
marital status	32.5%	84.55%

TABLE IV
SUMMARY OF RESULTS

VI. Conclusion

In this paper, we propose an MLP-based system for inferring hidden attributes in OSNs in an automated way. Our method can handle both classification and regression task, and it uses a large amount of available information in the OSN itself, including friendship relationships and visible attributes of friends. We measured the performance of our inference system on a real OSN, and the results are promising: our method clearly outperforms naive predictors, and it achieves similar performance as prior work did but using less information (we do not use group membership information and very specific information sources, such as tags on photos and appearance in a story). Our future work is concerned with experimenting with other machine learning based approaches and using our inference system to design tools that help people preserving their privacy.

VII. Acknowledgement

We would like to thank Boldizsár Bencsáth for his valuable help in obtaining the data for the experimental part of our research.

References

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *In Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2002, pp. 57–66.

[2] J. He, W. W. Chu, and Z. (victor Liu, "Inferring privacy information from social networks," in *IEEE International Conference on Intelligence and Security Informatics*, 2006.

[3] C. Y. Johnson, "Project 'Gaydar': An MIT experiment raises new questions about online privacy," Sep. 2009. [Online]. Available: http://www.boston.com/bostonglobe/ideas/articles/2009/09/20/project_gaydar_an_mit_experiment_raises_new_questions_about_online_privacy/?page=full

[4] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Inferring private information using social network data," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 1145–1146. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526899

[5] M. Mo, D. Wang, B. Li, D. Hong, and I. King, "Exploit of online social networks with semi-supervised learning," in *IJCNN*, 2010, pp. 1–8.

[6] M. Mo and I. King, "Exploit of online social networks with community-based graph semi-supervised learning," in *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ser. ICONIP'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 669–678. [Online]. Available: http://dl.acm.org/citation.cfm?id=1939659.1939746

[7] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in *Proceedings of the third ACM international conference on Web search and data mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 251–260. [Online]. Available: http://doi.acm.org/10.1145/1718487.1718519

[8] K. Thomas, C. Grier, and D. M. Nicol, "unfriendly: multi-party privacy risks in social networks," in *Proceedings of the 10th international conference on Privacy enhancing technologies*, ser. PETS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 236–252. [Online]. Available: http://dl.acm.org/citation.cfm?id=1881151.1881165

[9] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 531–540. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526781

[10] B. Li and M.-H. Meng, "Computer-aided detection of bleeding regions for capsule endoscopy images," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 4, pp. 1032 –1039, april 2009.