Empowering users of social networks to assess their privacy risks

Empowering users of social networks to assess their privacy risks

Vladimir Estivill-Castro
Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona 08018 Spain
Email: vladimir.estivill@upf.edu

Peter Hough
Center for Research in Complex Systems
School of Computing and Mathematics
Charles Sturt University
Panorama Avenue
Bathurst NSW 2795 Australia
Email: p_hough1@yahoo.com.au

Md Zahidul Islam
Center for Research in Complex Systems
School of Computing and Mathematics
Charles Sturt University
Panorama Avenue
Bathurst NSW 2795 Australia
Email:zislam@csu.edu.au

***Abstract*—Millions of users place data about themselves on on-line social networks and, while probably they have an interest on some of this information to be publicly available, they certainly may consider some of this information shall remain confidential. Simultaneously, the data provides benefits as such data enables personalization which increases the quality of service; and thus, it is regularly analyzed with data mining techniques. Since privacy directly correlates to the control users have regarding the data about themselves, this paper provides a technique by which operators of on-line social networks can improve the service to their users by empowering the users to appraise the privacy risks that some information they provide results in others inferring confidential attributes.**

I. Introduction

The exponential growth of on-line social networks has resulted in big data collections of personal information. This provides an opportunity for service providers and others to identify valuable information such as social and political trends, emerging group opinions, and marketing opportunities. More importantly, it also enables organisations to capture how and when (and often where) people use Web applications. This poses challenges to privacy [1].

Big data technologies have only fueled more the debate about seriousness of the issue, and the media has been attracted by the lively discussion [2]. In fact, Web applications must carry out surveillance and store usage data to provide personalisation and high quality services. The current quality of service of many Web applications (and in particular search engines) is due to the analysis of usage data [3]. Data-mining techniques are regularly used to analyze these data sets for personalization and it is well established that personalisation conflicts with users' privacy [4]. However, users have little control over the analysis and usage of the data they generate. Since privacy is mostly about the control [5], [6] persons (and organisations) can impose on the data about themselves, it is critical to produce instruments so users can regulate the usages that their data can be subject to. Today, what constitutes potential breaches, or the risks that data may be subject to disclosure or compromise, is often difficult to explain to the person the data is about. Since privacy is now considered to involve both 1) the CONTROL [5] of access to data as well as 2) the PRACTICE of such control [6, Page 403], we propose a mechanism that enable users to evaluate the trade-offs in privacy with respect to risk of disclosure. We describe an algorithm to measure privacy and display feedback that will enable users to understand the trade-off between personalisation and privacy. That is, everyday users may not be familiar with risk management and potential attacks, but the tool suggested here provides meaningful feedback to allow users of on-line social networks, for instance, to control the trade-off between privacy-risk with personalization.

Developers of Web applications wish to provide a better service and reduce privacy concerns. Privacy awareness leads to suspicion [7], which subsequently leads to PRIVACY-ACTIVE BEHAVIOR (that is, behavior of the user to conceal information). Also, perceived privacy-risk had a strong negative influence on the extent to which respondents participated in online subscription and purchasing [7]. Thus, we propose techniques by which users may investigate the sensitivity of a piece of information about themselves (typically the value for an attribute column, but the technique clearly extends for relational information like connectedness to another user). Such attribute value will be referred here as the confidential attribute-value pair. The technique will inform the user of what other of the publicly available attributes contribute to the disclosure of such confidential attribute-value pair. The user may then conceal one or more of those other attributes that facilitate prediction of the confidential attribute-value pair, and repeat the exercise until satisfied that there is enough uncertainty that malicious data miners would not infer the confidential attribute-value pair with sufficient confidence.

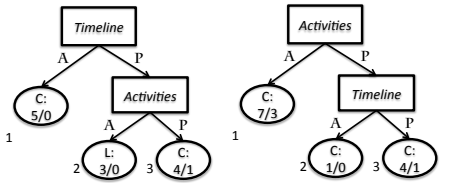
II. Concealing an attribute-value pair

To describe the proposed technique we assume that the adversary has assembled a large dataset T of information about users and is attempting to use classification methods to detect the confidential attribute-value pair. Thus, the data

Authors appear in alphabetical order as is customary in mathematics and theoretical computer science [see `en.wikipedia.org/wiki/Academic_authorship` and the American Mathematical Society *Culture Statement*]. All authors contributed intellectually to the scientific and technical ideas of the paper plus all are in a position to describe and present these ideas.

The first author would like to thank support from project TIN2013-49814-EXP of the MINISTERIO DE ECONOMíA Y COMPETITIVIDAD, SPAIN.

The third author would like to thank the Faculty of Business Compact Funding in Charles Sturt University, Australia.



(a)

(b)

UID	int
Family Info	{P, A}
Activities	{P, A}
Interested in	{M, F , B}
Timeline	{P, A}
Sentiment	{L, C}

(c)

Fig. 1: Two trees that result from the expansion from alternative roots because both attributes have an information gain larger than the threshold β.

has been organized as a large table with the last column as the dependent variable and the earlier columns as the independent variables. That is, the data is a set of vectors of the form $(\vec{x}, y)$, where $\vec{x} = (x_1, x_2, \ldots, x_d)$ are the d predictor attributes. The attribute y under investigation (the last column) is the one the user has the privacy preference to keep confidential, while the adversary's aim is to disclose it. For ease of discussion, Fig. 1c shows some columns (of the schema) of a sample fact-table. The first two attributes are *Family Information* and *Activities* (which can take two values: A for *absent* or P for *present*). The gender of the user's interests can have values M for *male*, F for *female*, and B for *both*. The fourth attribute is the *Timeline* which also can be A for *absent* or P for *present*. Finally, the last attribute records a sentiment as to whether the person feels *connected* (C) or *lonely* (L). We assume the database T is available as a training set (certainly, such data would be to the organization providing the on-line network service), and our technique uses T as input so users whose sentiment column is concealed can obtain feedback on the predictability of their personal attribute-value pair on the basis of the other attributes.

Decision-tree learning algorithms produce classification models that map attribute-value pairs in the independent columns $\vec{x}$ to a prediction of the dependent column y. The leaves are assigned the class labels and the tree structure encodes classification rules. Paths (from the root to a leaf) represent a conjunction of the attributes. Thus, decision-trees can be converted to Boolean rules. If $l_{y=b}$ is the label for the leaf on the path $x_{i_1} = a_{11}$, $x_{i_2} = a_{21}$, $\ldots$ $x_{i_p} = a_{p1}$, then the logic rule $(x_{i_1} = a_{11}) \wedge (x_{i_2} = a_{21}) \wedge \ldots \wedge (x_{i_p} = a_{p1}) \rightarrow y = b$ is the classification rule for this leaf. The disjunction of all the rules for which $y = b$ is the (disjunctive normal form - DNF) formula to classify the class $y = b$. Fig 1a and Fig. 1b show two trees for the data set having a schema as Fig. 1c. Heuristically, algorithms that construct a tree from a training set use a measure of classification power of the attributes to greedily build the trees top-down [8]. Since generalization power (and thus accuracy in unseen examples) is proportional to how shallow is the tree, the attribute to be chosen at the root of each subtree that is being expanded ought to be the most relevant (the so called best split) [9].

The *information gain* is usually a good measure for deciding the relevance of an attribute. The *expected information gain* $IG(T, a)$ measures the change in *information entropy* from a prior state of knowledge to a state where some information is given. Let H denote the information entropy (which is a measure of uncertainty), then $IG(T, a) = H(T) - H(T|a)$. That is, an adversary (that knows the data set T) gains certainty when additionally learning a about a user. To recall the formal definition, let x_i be one of the independent attributes and assume it is a discrete attribute with domain D_i, then

$$IG(T, x_i) \quad (1)$$
$$= \quad H(T) - \sum_{v \in D_i} \frac{|\{\vec{x} \in T | x_i = v\}|}{|T|} \cdot H(\{\vec{x} \in T | x_i = v\}).$$

Thus, our technique consist of generating a large forest of decision-trees. For simplicity, we assume all attributes are discrete with few values in each domain (this just ensures once an attribute appears in a path, it will not appear further down). The forest is generated taking into consideration a particular user u willing to conceal that $y = b_u$. Let $\beta \geq 0$ be a sensitivity parameter. The instrument providing a sense of privacy-risk to such user generates a decision-tree using the standard top-down algorithm. At each node T_i under expansion, we evaluate the information gain of each attribute. We create a list at the node T_i of all attributes whose information gain was larger than β. While the standard decision-tree building approach obtains only one tree (typically interested in only one classification model), we actually branch into a new tree for every attribute in the list for T_i. Ignore, for the moment, any pruning or alternative approach to avoid over-fitting.

Once all trees of this form are constructed, we look at all paths in all these trees where the leaf is labeled $y = b_u$ and user u would land in such leaf. This provides a large list $R(b_u, \beta) = \{R_1, R_2, \ldots, R_m\}$ of classification rules[1]. Now, for each of these rules R_j, we can define the support s_j (the ratio of the number $|R_j|$ of records that fall into the leaf of the rule to the total $|T|$ of records in the data set); that is, $s_j = |R_j|/|T|$. Similarly, the confidence c_j of the rule R_j is the ration of the number $|R_j^+|$ of records correctly classified to the number $|R_j|$ of records in the leaf. The support is an estimate of the probability that the rule is applicable to user u, while the confidence is an estimate of the probability the rule would reveal/compromise the attribute-value pair of u. User u is interested in ensuring the adversary can not find rules with high support and high confidence. In other words, the user is interested in blocking rules where the number of bits of information about the confidential attribute-value pair is high. The safety $-log(c_j) - log(s_j)$ is the number of bits needed by the adversary to disclose the confidential attribute pair after the rule R_j is applied. Alternatively, we can define the sensitivity S_j of a rule R_j as $S_j = c_j + s_j = |R_j|/|T| + |R_j^+|/|R_j|$.

The attribute that provides the highest information gain is the one user u should be removing/blocking/modifying in order to protect the confidential attribute-value pair. Recall, however, that using an information measure to expand one node at a time is a greedy heuristic, and that there may be another attribute that (in combination with a few others) results in highly sensitive rules. Thus the aim is to provide feedback to the user about the attributes that participate in as many as possible of the high sensitive rules. We present two approaches to identify these attributes.

[1]Because the $\wedge$ (Boolean AND operator) commutes, a rule may appear in more than one of the trees of the forest, but only one is kept. For example, this happens in the trees of Fig. 1.

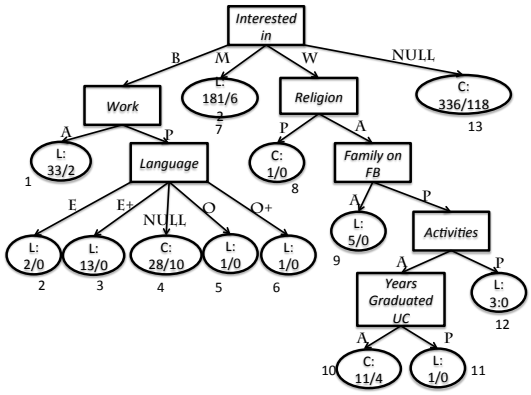


Fig. 2: A decision-tree with 13 leaves leading to 13 rules.

III. Informing the user

Again, given a users selection of confidential attribute and risk threshold β, the set $R(b_u, \beta)$ of sensitive rules (those rules R_j whose sensitivity S_j is larger that β) defines a set of sensitive attributes; namely, all those attributes that appear in the antecedents of the sensitive rules. Formally, if x_i =? appears in the antecedent of some sensitive rule $R_j \in R(b_u, \beta)$, then $i \in \{1, \ldots, d\}$ is the index of a sensitive attribute. The cumulative sensitivity CUM_SENSITIVITY of an attribute x_i is the sum of the sensitivities of the rules the attribute is an antecedent for. Namely

$$\text{CUM\_SENSITIVITY}(x_i) = \sum_{R_j \in R(b_u,\beta) \wedge x_i \text{ is an antecedent of } R_j} S_j.$$

Sorting all attributes in descending order of cumulative sensitivity is the output of our first algorithm. It suggests to the user what attributes to modify/conceal in order to reduce the risk of disclosure of the attribute-pair the user considers confidential.

A second algorithm TOTAL_COUNT just counts the number of times an attribute x_i ($i \in \{1, \ldots, d\}$) appears among the sensitive rules.

$$\text{TOTAL\_COUNT}(x_i) = \|\{R_j \in R(b_u, \beta) \mid x_i \text{ is an antecedent in } R_j\}\|.$$

The attributes are presented to the user in descending order of TOTAL_COUNT(x_i) value. Heuristicly, the suggestion by CUM_SENSITIVITY will minimize the attributes that need to be concealed in order to maximize protection to sensitivity. On the other hand, TOTAL_COUNT will minimize attributes than need to be concealed to eliminate as much as possible the inference rules.

These indicators of the most influential attribute in the perspective of user u enable to present to u a ranking of his/her public attributes to conceal in order to reduce the ability of others to infer the confidential value b_u.

IV. A first experiment

We now contrast the CUM_SENSITIVITY heuristic and the TOTAL_COUNT heuristic with a straw-man proposal of simply pseudo-randomly concealing an attribute in the dependent-variable set (this approach is part of NOYB [10]). For this

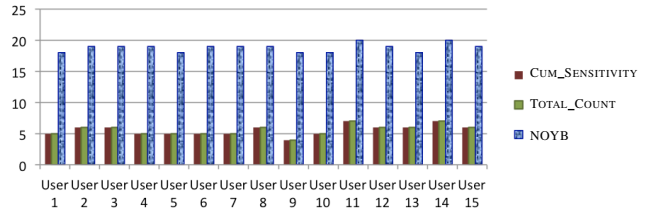


Fig. 3: Number of repetitions (and dependent attribute suppressions) until all sensitive rules are eliminated.

we use a dataset from *Facebook*. The data set has 615 users each with 25 attributes as independent variables and one class attribute (the *sentiment*) with values *lonely* or *connected*. We used 600 users for the generation of the forest of trees and a random subset of 15 users as the ones that later desire to conceal their sentiment. Fig. 2 shows one possible decision-tree learned in the generation of the forest. From the forest, we find that a total of 16 rules are sensitive rules.

We set uniformly the value $\beta = 1.0$. A rule is obtained by following a path from the root to a leaf in a trees in the forest. This bounds the rules applicable to a user to the number of trees in the forest, but there may be less rules triggered if their sensitivity is below the threshold β.

We assume that each of the privacy-concerned users applies CUM_SENSITIVITY, TOTAL_COUNT or a random choice. They iterate the following steps from the top attribute in the ranking and

1) identify the most influential attribute, and suppress it, modify it or whatever other option the environment enables to remove the availability of this independent variable value, and
2) continue with the next attribute in the ranking

until the sensitivity is down to zero (that is, no more sensitive rules appear). Each user adheres to one of the heuristics CUM_SENSITIVITY, TOTAL_COUNT, or a random choice for all of the repetitions of the iteration. Fig. 3 shows plots of the number of iterations necessary to eliminate all sensitive rules. It illustrate that CUM_SENSITIVITY and TOTAL_COUNT require between 4 to 8 iterations (and thus only 4 to 8 attributes are concealed); but the average number of randomly suppressed independent attributes is above 17. In fact for CUM_SENSITIVITY and TOTAL_COUNT the number of attributes suppressed across all users is never more than 8, and the average is 5. The effectiveness of the heuristic CUM_SENSITIVITY and TOTAL_COUNT can also be observed in Fig. 4 where we have plotted the cumulative sensitivity after one, two and three attributes are eliminated (we use NOYB and zero iterations, as a point of reference). It is also not hard to observe that, as expected, since CUM_SENSITIVITY is a more informed heuristic, it performs better than TOTAL_COUNT over the 15 users. Although after 3 iterations both have removed 75% of sensitive rules, for the first two iterations, CUM_SENSITIVITY removes more sensitive rules and reduces the cumulative sensitivity more than TOTAL_COUNT across all 15 users.

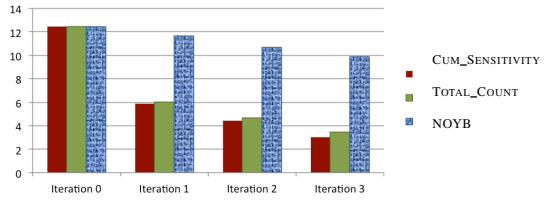


Fig. 4: Average sensitivity (across 15 users) of the attribute (*sentiment*, C) decreases much more rapidly for CUM_SENSITIVITY and TOTAL_COUNT as opposed to NOYB.

V. THEORETICAL FOUNDATIONS

Our method resembles techniques for attribute selection [11, Section 7.1], where the aim is to obtain the most succinct subset X' of the independent variables X in order to build accurate classifiers [12]. We use such fundamental approach [13] to provide a theoretical justification for our approach, while at the same time highlighting the key differences. By the mere presence of the available data set T, the adversary has an estimate to $Pr(y = b|X = \vec{x})$, where $\vec{x}$ is the record for a given user. In feature-space reduction, the goal is to obtain a subset X' so that, when we let $\vec{x}_{X'}$ be the vector for $\vec{x}$ projected into the columns in X', the value $Pr(y = b|X' = \vec{x}_{X'})$ is as close as possible to $Pr(y = b|X = \vec{x})$, for all users $\vec{x}$. Here we observe the first difference. The aim here is not to provide global feedback for all users jointly, but feedback tailored to each given user. Of course, if some attribute is a strong predictor this is unavoidable (since *pregnant* determines *female*, then all female users who wish to conceal their gender, will receive advice to conceal whether they are pregnant or not, but even in this case, since this is tailored to the attribute-value pair, the advice may only be provided to those actually pregnant).

In feature selection, using $Pr(y = b|X' = \vec{x}_{X'})$ as a representation as close as possible to $Pr(y = b|X = \vec{x})$ is directly evaluated by the information-theoretic measure of cross-entropy [13] when interpreted as the Kullback-Leibler divergence (or KL-divergence)

$$\delta_{X'}(\vec{x}) = D[Pr(y = b|X = \vec{x}), Pr(y = b|X' = \vec{x}_{X'})]. \quad (2)$$

In this case, $Pr(y = b|X = \vec{x})$ is the true distribution and $Pr(y = b|X' = \vec{x}_{X'})$ is the approximating distribution. Since the goal is to capture classification well for all $\vec{x}$ globally, the task for feature selection is to find X' such that $\Delta(X') = \sum_{\vec{x}} Pr(X = \vec{x}) \cdot \delta_{X'}(\vec{x})$ is minimum. There are certain practical challenges why computing such optimal set X' of features is only performed heuristically by a backward (greedy) method that starts with the set $X'_0 = X$ of all features and progressively eliminates attributes.

In the case of privacy-protection, the requirement would be the reverse. We aim to maximize the error of representation when the true $Pr(y = b|X = \vec{x})$ distribution is represented by $Pr(y = b|X' = \vec{x}_{X'})$. And clearly, we want a forward approach [14]. Recall that a forward approach in feature selection starts with $X' = \emptyset$ and repeatedly adds the most relevant column, which in this setting corresponds to the feature x_i that maximizes the expected cross-entropy between $Pr(y = b|X' = \vec{x}_{X'})$ and $Pr(y = b|X' = \vec{x}_{X' \cup x_i})$. But, we tailor with respect to the user who demands feedback, so we do not take the expectation over all users, so we shall not use $\Delta(X')$. But we do take into account the confidence of the rule for the applicability of the rule (the term $Pr(y = b|X = \vec{x})$).

Moreover, there is another fundamental reason why, in our case, the forward approach is required. In the heuristic using correlations to approximate Markov blankets (in it self an approximation the optimization of Equation (2) [13]) that backward method removes redundant independent variables even if they are good predictors. However, for privacy-protection, if two independent variables x_i and x_j are both good predictors of the independent variable y, we must remove both. This also justifies why one decisions-tree is insufficient and we do the forest exploration. Moreover, the literature on feature-selection has pointed out that a single decision-tree for identifying features suffers from data fragmentation [15]. Our extraction of all significant rules avoids this.

A. Personalization

We first emphasize that the challenge here is personalized. Again, for the moment consider the situation of a training set T of existing users and that user u is about to join the social network. By analyzing the training set T without any information about u, we can obtain which of the d attributes is the most relevant by the expected information gain $IG(T, x_i)$defined in Equation (1). Let x_{IG} be the attribute that maximizes $IG(T, x_i)$ among all attributes x_i. Thus, the attribute x_{IG} is chosen as the attribute at the root of a decision tree built using expected information gain.

We now present a specific scenario that illustrates the need for personalization. Suppose that the domain of the confidential attribute y has $|D_y| > 2$ discrete values. User u makes public only one attribute x_i that is different from x_{IG}. But, suppose the data is such that if we were to build a one level decision tree with x_i at the root, we find that we get $|D_i|$ branches and that the leaf for u is pure with all cases from T at this leaf also having value d_u. The other leaves have balanced impurity with the cases from T landing with equal frequency among the other values in D_y. In this case $IG(T, x_i)$ is likely to be small, and well beyond $IG(T, x_{IG})$. That is, x_i is not an important attribute in the context of T for predicting y. However, from the personal point of view of user u's privacy, visibility of attribute x_i is disastrous as it perfectly predicts the confidential attribute $y = d_u$ (the attribute-value) pair. User u does not care that x_i is bad at discriminating other values of the domain D_y, it is a risk to users u's privacy that x_i predicts perfectly the attribute-value pair that corresponds to u.

Thus, in what follows, we discuss the expected information gain measure $IG(T, x_i)$ in a *personalized* version to a particular user u, and denote that as $IG_u(T, x_i)$. Given the user u (and recall we denote as $d_u \in D_y$ the confidential value for attribute y that u considers confidential), we collapse the classes of the confidential attribute y into two classes; namely, one class is $y = d_u$ while the second class is $y \neq d_u$. Consequently, the definition of the measure $IG_u(T, x_i)$ is simply $IG_u(T, x_i) = H_u(T) - H_u(T|x_i)$ where now H_u indicates that we have a distribution where the random variable is not only discrete, but it can only take two values (it is a Bernoulli random variable). If we then build a decision-tree using the measure $IG_u(T, x_i)$ to chose what attribute to

UID	int
Family Info	{ P, A }
{ *Activities Interested in* }	{ (P,M), (P,F), (P,B), (A,M), (A,F), (A,B), }
Timeline	{ P, A }
Sentiment	{ L, C }

(a)

UID	int
Family Info	{ P, A }
{ *Activities Interested in* }	{ (P,F), *Activities*≠P ∨ *Interested in*≠F }
Timeline	{ P, A }
Sentiment	{ *Sentiment*= L, *Sentiment*≠L }

(b)

Fig. 5: Composite attributes and personalization define views on the training set T.

place at the root, the attribute that determines u's confidential attribute will come on top.

B. Formulation of the privacy-risk protection problem

A set S of attributes with $S = \{x_{s1}, x_{s2}, \ldots, x_{s|S|}\}$ can be interpreted as one attribute with domain $D_S = D_{s1} \times D_{s2} \times \ldots D_{s|S|}$.This is simply a iso-morphic view on the training set T. For example, in the schema of our running example from Fig. 1c, we could consider the two attributes *Activities* and *Interested in* as a composite attribute $S = \{Activities, Interested\ in\}$, then the schema becomes as Fig. 5a. Moreover, if user u makes public the set S of attributes, then we also consider a view of the training set personalized to the attribute-value pairs that u releases. We not only build a view where S is consider a composite attribute, but also the domain values are collapsed as those that match u's vector of attribute-value pairs for S and those which do not. For example, if we personalize the schema of Fig. 1c to a user who releases his/her values (P,F) for *Activities* and *Interested in*, and his confidential attribute is *Sentiment* value L, then the schema is as Fig. 5b. With the understanding from the two previous subsections we can formally state the problem.

Problem 1: Given a training set T and the data-vector of a new user u, (and a threshold $\theta_{\max}$ measured in bits for when the confidential attribute is considered too likely to be determined, and a threshold $\theta_{\min}$ measured in bits for when the confidential attribute is considered to be safe) find the smallest set S of attributes so that, for all subsets of attributes S_o (with $S \cap S_o = \emptyset$, but S_o could be empty), if $IG_u(T, S \cup S_o) \geq \theta_{\max}$, then $IG_u(T, S_o) \leq \theta_{\min}$.

We note that $IG_u(T, U)$, for a subset U of attributes of T represents the evaluation of IG on the view of T collapsed for the values of user u in the set U and the class attribute $y = b_u$.

First, we make an observation regarding the formal statement of the problem. If there is no subset S of the attributes so that $IG_u(T, S) \geq \theta_{\max}$, then the solution is empty. This is the case when even knowledge of all the attributes of user u is insufficient to gain more information that the threshold $\theta_{\max}$. Simply put, the data set T and knowledge of u can not construct a classifier for $y = d_u$ versus $y \neq d_u$ that is sufficiently accurate. Secondly, the definition makes S the exact set of attributes that causes the attribute-value pair to be at risk, and also ensures that any other combination is safe once the attributes in S are excluded.

This formalization now clearly spells out the difference with the objectives of feature selection or attribute selection. Moreover, the literature has deeply investigated measures of attribute relevance. The expected information gain IG (also named the *transmitted information* [16]), is popular to evaluate attribute relevance, but other well studied measures include (1) information-based measures (*gain ratio*, *distance measure* [16]) or (2) probability-based measures [16] as well as (3) permutation-importance measures [17]. The main concern on varying which measure to use (when growing a decision tree) is the bias for attributes with large domains.

The formalization of the problem presented in Problem 1 shall not pose a much different problem if the information gain measure is swapped by one of the others. The reason being that the views we construct actually collapse domains into discrete domains of two values (matching the data in u's record or not).

Now, the challenge of searching for the set S as per Problem 1 suffers the same difficulties as the formulations of feature selection in the literature. For the large datasets involved in on-line social media, we simply cannot afford to test all subsets S of attributes. However, CUM_SENSITIVITY and TOTAL_COUNT can be used to create a ranking and allow users to iteratively approximate the S of Problem 1 from $S = \emptyset$, build a set S' that fits their needs (analogously to forward approach [14] of feature selection discussed earlier).

We note that there is significant literature in the search of alternative classifiers; that is, researchers have investigated obtaining more than one classification tree, or obtaining classifiers that use other attributes while maintaining accuracy. The main reasons is to use attributes that are less costly to evaluate [18], to have alternatives in case some attributes are missing on a particular instance [19], or to create ensemble of classifiers and improve accuracy [20], [21]. A technique called "Cascading and Sharing Trees (CS4)" [22, and references] generates forest of decision trees even when the root may be a very weak predictor on its own. Seeking alternative decision trees also explore the possibility of finding sets of attributes, that jointly are very strong predictors although each attribute on its own is not informative [23]. Our heuristics, CUM_SENSITIVITY and TOTAL_COUNT, heuristics are inspired by this later approach.

C. Experimental validation

Naturally, the question is, how close is an answer built one attribute at a time by the heuristics CUM_SENSITIVITY and TOTAL_COUNT to the optimal solution as per Problem 1. In this section we provide results of an experimental validation on the results of CUM_SENSITIVITY and TOTAL_COUNT. In particular, we use the freely distributed dataset `infer-attrib/SEP4.txt` previously used [24] to infer attributes of users in a social-network. This data set has over 5,000 users and there are around 10,000 attributes possibly present for a user; although on average, each user has about 4 attributes. We investigated the privacy of those attributes that appear at least on 60 users and in no more than

200 users. For each of these 19 attributes, we extracted users that have the attribute present and computed by exhaustive search the set S of size 3 that uses other attributes and provide jointly the largest information gain (that is, we computed the optimal S in Problem 1). We then also applied three iterations of each of CUM_SENSITIVITY and TOTAL_COUNT, obtaining with each an approximation to S.

We have clustered all 437 users into user-pattern, since some users have exactly the same attributes (each of the users in a user-pattern results in the same output, as all their attributes match). Across a user-pattern, the result vary since all methods (CUM_SENSITIVITY, TOTAL_COUNT and the exhaustive search) depend on the attribute-values of the user who is attempting to reduce the risk of disclosure of its sensitive attribute. In general, both CUM_SENSITIVITY and TOTAL_COUNT find at least 2 attributes in common with the optimal set of 3 attributes (for attribute 542 they only find 1, but for other like 564, 322 and 79, the heuristic finds all attributes in the optimum set).

Note that the heuristics are very strong in approximating the optimum. Also, heuristics CUM_SENSITIVITY and TOTAL_COUNT work effectively as disrupter of inferences. Note that the potential difficult case for CUM_SENSITIVITY and TOTAL_COUNT is when the optimum set S consist of 3 attributes that together are very effective in predicting the sensitive attribute but each attribute individually does not do so well. If CUM_SENSITIVITY and TOTAL_COUNT remove 2 out of the 3 attributes in their suggestion, and leave one attribute from the optimum set, this is not a problem because this attribute on its own is a bad predictor.

The heuristics work well when users does not know ahead of time how many attributes they wish to remove to protect the sensitive attribute. The heuristic enables to start and conceal one attribute after another until the desired risk-minimization has been achieved. We believe this is user-friendly in the sense that it would be more manageable by users than forcing users to anticipate how many attributes to conceal.

VI. CONCLUSIONS

We have provided two heuristics and shown that they could be applied interactively by users to incrementally occlude attributes and properties about themselves and reduce the exposure and risk of a sensitive attribute (that of course is not made public). This heuristics perform very well on a real-data sets and have significant justification for why they should be effective.

Our approach here enables the users themselves to evaluate their risk and and also the predictability of sensitive attributes by other information they are supplying. We consider this a very important initial step as the emergence of big-data will soon collect even more attributes about individuals; thus it is important that those the data is about can assess whether such information is enabling others to predict attributes they consider sensitive.

REFERENCES

[1] J. M. Such, A. García-Fornes, A. Espinosa, and J. Bellver, "Magentix2: A privacy-enhancing agent platform," *Eng. Appl. of AI*, vol. 26, no. 1, pp. 96–109, 2013.

[2] B. Winterford "Academics get personal over big data" July 11th, 2014 www.itnews.com.au/News/389522,academics-get-personal-over-big-data.aspx

[3] R. A. Baeza-Yates and Y. Maarek, "Usage data in web search: Benefits and limitations," in *Scientific and Statistical Database Management - 24th Int. Conf., SSDBM 2012*, LNCS, vol. 7338. Chania, Greece: Springer, pp. 495–506.

[4] N. Ramakrishnan, B. Keller, B. J. Mirza, A. Grama, and G. Karypis, "Privacy risks in recommender systems," *IEEE Internet Computing*, vol. 5, no. 6, pp. 54–62, 2001.

[5] R. Wishart, R. Corapi, A. Madhavapeddy, and M. Sloman, "Privacy butler: A personal privacy rights manager for online presence," in *Eigth Annual IEEE Int. Conf. on Pervasive Computing and Communications, PerCom*. Mannheim, Germany: IEEE, 2010, pp. 672–677.

[6] G. Gürses and B. Berendt, "The social web and privacy: Practices, reciprocity and conflict detection in social networks," in *Privacy-Aware Knowledge Discovery, Novel Applications and New Techniques*, F. Bonchi and E. Ferrari, Eds. CRC Press, 2010, pp. 395–429.

[7] J. Drennan, G. Sullivan, and J. Previte, "Privacy, risk perception, and expert online behavior: An exploratory study of household end users," *JOEUC*, vol. 18, no. 1, pp. 1–22, 2006.

[8] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[9] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 35, no. 4, pp. 476–487, 2005.

[10] S. Guha, K. Tang, and P. Francis, "NOYB: privacy in online social networks," in *Proc. first workshop on Online social networks*, WOSN '08. NY, USA: ACM, 2008, pp. 49–54.

[11] I. H. Witten and E. Frank, *Data Mining*. Morgan Kaufmann, 1999.

[12] B. Chizi, L. Rokach, and O. Maimon, "A survey of feature selection techniques," in *Encyclopedia of Data Warehousing and Mining, (4 Volumes)*, 2nd ed., J. Wang, Ed. IGI Global, 2009, pp. 1888–1895.

[13] D. Koller and M. Sahami, "Toward optimal feature selection," in *Machine Learning, Proc. Thirteenth Int. Conf. (ICML '96)*, Bari, Italy: Morgan Kaufmann, 1996, pp. 284–292.

[14] M. Singh and G. M. Provan, "Efficient learning of selective bayesian network classifiers," in *Machine Learning, Proc. Thirteenth Int. Conf. (ICML '96)*, Bari, Italy: Morgan Kaufmann, 1996, pp. 453–461.

[15] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1-2, pp. 273–324, 1997.

[16] A. P. White and W. Liu, "Bias in information-based measures in decision tree induction," *Machine Learning*, vol. 15, no. 3, pp. 321–329, 1994.

[17] H. Deng, R. G. C., and E. Tuv, "Bias of importance measures for multi-valued attributes and solutions," in *Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st Int. Conf.*, LNCS, vol. 6792. Espoo, Finland: Springer, 2011, pp. 293–300.

[18] P. D. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *J. Artif. Int. Res.*, vol. 2, no. 1, pp. 369–409, Apr. 1995.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] M. Amasyali and O. Ersoy, "Classifier ensembles with the extended space forest," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 3, pp. 549–562, March 2014.

[21] J. Rodriguez, L. Kuncheva, and C. Alonso, "Rotation forest: A new classifier ensemble method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1619–1630, Oct 2006.

[22] J. Li and K. Ramamohanarao, "A tree-based approach to the discovery of diagnostic biomarkers for ovarian cancer," *8th Pacific-Asia Conf., PAKDD*, LNCS, vol. 3056. Sydney: Springer, 2004, pp. 682–691.

[23] M. Z. Islam and H. Giggins, "Knowledge discovery through sysfor: A systematically developed forest of multiple decision trees," *Proc. 9th Australasian Data Mining Conf. CRPIT 121*, ACS, 2011, pp. 195–204.

[24] N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, p. 27, 2014.