Understanding Site-Based Inference Potential for Identifying Hidden Attributes

W. Brad Moore Yifang Wei Adam Orshefsky Micah Sherr Lisa Singh Hui Yang
Georgetown University
Washington, DC 20057

***Abstract*—The popularity of social networking sites has led to the creation of massive online databases containing (potentially sensitive) personal information, portions of which are often publicly accessible. Although most popular social networking sites allow users to customize the degree to which their information is publicly exposed, the disclosure of even a small, seemingly innocuous set of profile attributes may be sufficient to infer a surprisingly revealing set of attribute-value pairings. This paper analyzes the predictive accuracy of existing and ensemble inference algorithms to infer hidden attributes using publicly exposed attribute-values. For our tested population, we find that (i) certain attributes are more accurately predicted than others, (ii) each tested inference algorithm is well-suited for inferring a particular subset of attributes, and (iii) these subsets of inferable attributes often have little overlap. Taken collectively, our results indicate that the amount of information one can extract from a given user's public profile is often greater than the sum of the attributes that the user has chosen to publish.**

I. Introduction

Online social networking sites provide a convenient and user-friendly method of sharing user-generated content. To participate in these online networks, a user creates a *profile* that may contain potentially sensitive information (e.g., status as a job seeker, relationship status, sexual orientation, etc.). Fortunately, most popular social networking sites also provide access controls over online profiles—for example, permitting users to reveal certain attribute-values only to authorized peers.

However, many users are not aware that their private data can potentially be revealed by straightforward analysis. In this paper, we study the *unintended leakage* of potentially sensitive information through users' publicly accessible online profiles. Specifically, we investigate the degree to which publicly available site-level data can be leveraged to determine undisclosed attributes of users. Our goal is to understand which attributes are most easily inferable (high leakage risk) and most useful for inferring hidden attributes (high inference potential).

Given the sometimes sensitive nature of information posted on social networking sites, a growing body of research attempts to understand and mitigate the risks associated with disclosing potentially sensitive information (cf. [5, 6, 13, 16, 19, 20]). Inferring attribute-values of individual profiles [4, 5, 14, 17, 19, 21] and reidentification studies [5, 6, 13, 16, 19, 20] are also growing areas of research.

Similar to some of these works, this paper demonstrates that despite existing privacy mechanisms, potentially sensitive information about a targeted user can be determined using other users' publicly accessible profile information. We exploit the observation that frequent patterns of a *site's subpopulation* or individuals with similar attribute-values on a site can be used to infer a particular user's hidden attribute-values.

Using a corpus of nearly 180,000 public profiles from LinkedIn and Google+ collected over several months, we construct a site-level inference engine using a combination of multi-attribute association rule mining, Latent Dirichlet Allocation (LDA) [3], and Naïve Bayes. Empirical evaluations using LinkedIn and Google+ show that hidden attributes can be inferred with high accuracy for some of our attributes. Our results indicate that because a segment of the population is not concealing sensitive attributes, deciding to merely conceal sensitive attributes from one's own online profile is insufficient to guarantee privacy. The disclosure of even a small, seemingly innocuous set of profile attributes is sufficient to infer a revealing set of attribute-value pairings.

This paper makes the following contributions: (1) A methodology for inferring sensitive attribute-values on online social media sites using a random site-based population; (2) An approach for developing a site-based inference engine using multiple inference algorithms, two of which are new; (3) A formal definition of the attacker model for the problem of inferring hidden attributes on social media sites; (4) The evaluation of our inference engine using two popular social media sites; and (5) An analysis of nearly 180,000 public profiles, highlighting trends on real-world social media sites.

II. Related Work

Previous work has demonstrated that online users publicly expose a significant amount of personal data on sites such as Facebook [9], including sensitive fields such as birth date, hometown, current residence, and phone number. While recent work has shown that Facebook users are reducing the amount of information they share publicly [6], other work by Chaabane et al. [4] of 100,000 Facebook users found that 75% revealed gender, 57% revealed interests, and 23% revealed their current city. While these attributes may seem less sensitive, they can be used to infer more sensitive attribute-values, as we demonstrate in Section V.

A number of researchers have proposed approaches for inferring sensitive attributes from online social networking sites. Zheleva and Getoor [21] use link-based classification to study the impact of friend attributes on the privacy of users by using the attribute-values of friends in common groups to infer a particular user's attribute-value. Crandall et al. [5] infer social ties using geographic proximity between two Flickr users. They find that a very small number of co-occurrences near each other in a short period substantially increases the

probability that the two people have a social tie. Kosinski et al. [12] apply statistical regression models to Facebook "Likes" to predict sensitive attributes, including political views and sexual orientation. Mislove et al. [17] use community detection metrics to infer attributes in two Facebook data sets. After identifying the community of the user, the authors determine the strength of the community using affinity and also consider the common attributes of the user community using modularity. Including friendship network structure into our inference engine is left for future work.

Chaabane et al. [4] use a Latent Dirichlet Allocation generative model [3] to identify relationships between different interests specified by users. The authors then map the interest to topic groups, and infer sensitive attributes using this topic structure. They show that Facebook users who are interested in similar topics with similar likelihoods have similar profile data. While components of their methodology are similar to ours, our work considers multiple attributes and multiple algorithms within a single prediction. One of the methods we consider here is an extension of the one proposed by Chaabane et al.—we use multiple attributes (as opposed to one) as evidence for an inference and apply the method to infer a wider range of attributes. Another inference study by Lindamood et al. [14] uses an extended Naïve Bayes classifier to infer political affiliations based on friendship links and user attributes for 35,000 Facebook profiles. Because we do not have link structure in our data set, we consider a variation of a Naïve Bayes classifier for site-based inference. Our variation, described in Section IV, generates multiple predictions using different numbers of attributes as evidence when determining the hidden attribute and then considers a majority vote of the results to find the final, hidden attribute-value.

Recent work in re-identification focuses on mapping records in different data sets to the same real world entity [8, 18, 19]. These works differs from ours since they attempt to re-identify individuals across data sets on different sites and do not explicitly build an inference engine for predicting sensitive attributes. Other recent work investigates methods to measure a user's susceptibility to an attacker and to protect privacy within a social network. Lie and Terzi [15] calculate privacy scores for profiles. Baden et al. [2] consider mitigating privacy risks by creating a social network, Persona, with users' privacy controls as a primary goal. The Diaspora social networking service aims to improve privacy through a distributed and community-oriented design, preventing a single organization from collecting sensitive personal information [7]. While these works are important directions, we focus on quantifying the privacy risks due to participating in *existing* social networks.

III. System and Attacker Models

We model a *site* as a data set $D(\underline{\text{ID}}, A_1, A_2, \ldots, A_m)$ containing $m+1$ attributes, where the primary key ID uniquely identifies a user of the site. The data set is a collection of user *profiles* (also called *records*). Conceptually, a profile contains demographic and other information about a user, represented as attribute-value pairings. Each attribute-value may be either a singleton (e.g., 35 in the case of age) or may be an unordered set of values (e.g., {HTML, Social Media, Budgeting} in the case of skills). Each record $\langle id^i, v_1^i, v_2^i, \ldots, v_m^i \rangle$ in the data set D corresponds to the i^{th} user's profile. We denote the profile for user i as $\mathcal{P}^i$ and use the notation $\mathcal{P}_j^i$ for the value of attribute A_j for user i. For simplicity, we assume that a user's profile is *sound* but may not necessarily be *complete* — that is, each attribute-value $\mathcal{P}_j^i$ either correctly describes user i or is null ($\perp$).

Our system model envisions three principals: the site operator, the accessor, and the user. The *site operator* maintains the data set and has access to all attribute-values in D. The *accessor* accesses a subset (i.e., a view) $V \subseteq D$ of the data set, usually through a web interface or an API. In practice, the accessor's view is often read-only and is limited to a number of profiles and attributes[1]. As discussed below, the adversary is a restricted instance of an accessor. Finally, a registered *user* of the site has a profile stored in D. In addition to the capabilities of an accessor, the user typically also has read-write access to many of the attribute-values in his/her profile.

The data set D may contain sensitive information. We assume that the site provides a permissions system in which users can define who has access to particular attributes in their profiles. For example, a user may restrict certain attributes to peer users (i.e., its "friends") while exposing other attributes to the public (and consequently, all accessors). The site may support fine- or coarse-grain access controls, and may not allow users to specify permissions for all attributes. At the extreme, the site may not allow the user to configure permissions for any of his/her attributes Without loss of generality, we define the function $\text{restricted}(i, A_j)$ to be $\top$ if user i restricts access to A_j to a subset of peer users of the service, and $\perp$ otherwise.

With the above definitions, we now more formally define site profiles:

Definition 1 (Profile): Given data set $D(\underline{\text{ID}}, A_1, A_2, \ldots, A_m)$, the *profile* for user i is $\mathcal{P}^i = \sigma_{\text{ID}=i}(D)$, where σ is the selection operator.

Definition 2 (Public Profile): Given a data set $D(\underline{\text{ID}}, A_1, A_2, \ldots, A_m)$, the *public profile* for user i is $P^i = \pi_{pub(A_j)}(\sigma_{\text{ID}=i}(D))$, where σ is the selection operator, π is the projection operator, and $\text{pub}(A_j) = \{A_j \in \mathcal{P}^i : \text{restricted}(i, A_j) = \perp \ \wedge \mathcal{P}_j^i \neq \perp\}$.

Note that, by Definitions 1 and 2, an accessor can access all attribute-value pairs in P^i.

Our model is intentionally generic and captures a variety of existing online services, including social and professional networking (Facebook, Google+, LinkedIn, Renren), microblogging (Twitter), dating (Match.com), social media sites (YouTube, Vimeo), job search (Monster), and third-party information providers (WhitePages, Spokeo), among many others. We do not distinguish between profiles that have been assembled by the user, e.g. during registration, and those that have been collected by a third party. In this paper, we demonstrate the effectiveness of our techniques on a social network (Google+) and a professional networking site (LinkedIn).

Attacker Model. We model the adversary as an accessor who wishes to learn information about a targeted user i and is not one of i's peers (that is, the adversary cannot access

[1]The accessor may itself be a registered user of the site. For example, this may be the case if outsiders have no access to D and only registered users can browse the site's profiles.

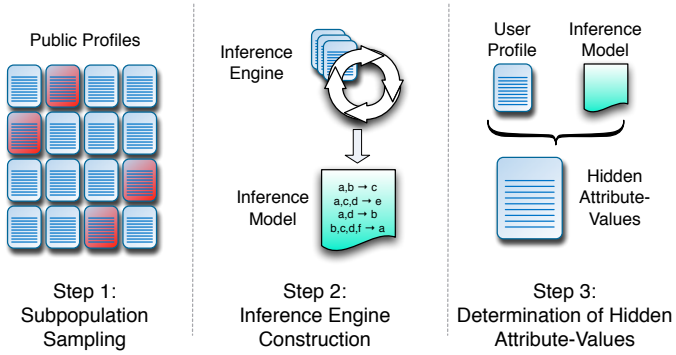

Fig. 1: Attribute inference methodology. The adversary samples a random subpopulation of site-level public profiles *(left)*, constructs site-level inference rules and models using the sampled public profiles *(center)*, and applies the inference engine to a targeted user's public profile to predict a hidden attribute-value *(right)*.

the value of attribute A_j when $\text{restricted}(i, A_j) = \top$). The adversary's goal is to discover information about i which is not in the public profile P^i.

To more formally define the attacker, we introduce a *truth function* $\text{truth}(i)$ that returns a set $\langle \overline{id}, \overline{v}_1^i, \overline{v}_2^i, \ldots, \overline{v}_m^i \rangle$ such that (1) $\overline{id} = \mathcal{P}_{ID}^i$, and (2) either $\overline{v}_j^i = \mathcal{P}_j^i$ if $\mathcal{P}_j^i \neq \perp$, or otherwise $\overline{v}_j^i$ is the correct value of attribute A_j for user i. Intuitively, $\text{truth}(i)$ is the complete set of correct values for user i for attributes $\text{ID}, A_1, \ldots, A_m$, and can contain values that are not in D. Hence, the adversary's goal is to infer the set $\text{truth}(i) \setminus P^i$. Note that this includes both attributes that are restricted using the site's permission system, as well as attributes that are unknown (i.e., null) to the site. Toward that end, in this paper we attempt to infer single attribute-values using data available to the adversary.

IV. Attribute Inference Methodology

Even though social networking sites often include privacy settings that allow a user to control which attributes in her profile are disclosed to the public, based on the previous literature presented in Section II, we make the observation that removing sensitive attributes from a public profile is insufficient to ensure that those attributes are not easily discoverable. In this paper, we are interested in understanding *how* a public attribute or public attribute combination can be used to infer hidden values. Therefore, we analyze how effective frequent patterns of a *site's subpopulation* are for inferring sensitive attributes that are hidden by a particular user.

We develop an attribute inference methodology for determining non-published attributes about a targeted user. Our methodology is based on the premise that an attacker may explore the site in question and then use this background knowledge to make inferences about a particular user's non-published attributes. Figure 1 shows the three steps in our high level attribute inference methodology: subpopulation sampling, inference engine construction, and determination of hidden attribute-values.

A. Subpopulation Sampling

The first step of our methodology is to randomly sample a subpopulation of profiles from a site containing a database D. More formally, our subpopulation D' has a representative sample of the attribute-value pairs of interest from D (i.e., $D' \subseteq D$). In practice, an adversary can trivially obtain a subpopulation sample by using a site's web interface or API.

B. Site-based Inference Engine Construction and Determination of Hidden Attribute-Values

There are many methods for building a site-based inference engine. We begin by extending two previously proposed approaches: one that uses Latent Dirichlet Allocation (LDA) [4] and another based on a modified Naïve Bayes method [14]. We then propose a new approach based on multi-attribute association rule mining. Finally, we consider an ensemble approach that incorporates all of the different techniques into the site-level inference engine. Construction of the inference engine is done offline and infrequently for a particular site; therefore, the cost of generating inference models or rules is not significantly burdensome to the adversary.

To clarify the different methods, we will use a toy example based on user data presented in Figure 2. In this example, D contains four attributes: id, gender, relationship, and industry. The adversary is interested in determining User 6's industry attribute-value. In this scenario, User 6 has decided to not make this attribute-value public. Using the site API, the adversary obtains D', a subset of D containing the public profiles for Users 1-5. The adversary will now generate his inference engine using these public profiles. The remainder of this subsection describes each of the methods that can serve as the basis for the inference engine that the adversary will build.

LDA Nearest Neighbor Inference. Chaabane et al. use the Latent Dirichlet Allocation (LDA) generative model to extract semantic links between *interest* attribute-values [3]. Our variation of their method is as follows.

Each profile $\langle id^k, v_1^k, v_2^k, \ldots, v_q^k \rangle$ in the subpopulation D' consists of the attribute-value pairs for some subset of attributes in D. We begin by considering a particular attribute A_q. Each attribute has a domain containing a set of values, $|A_q| = \{v_1, \ldots, v_m\}$. In LDA terms, we consider each attribute-value a word. For each attribute-value, v_k, we obtain its related Wikipedia categories to enhance the value sets. We first retrieve the top relevant article describing the attribute v_k from a free text index built by the Lemur Search Engine[2] over the entire Wikipedia stub contained in the ClueWeb09 collection[3]. The index's size is approximately 1GB for the compressed documents. Next, from each of these articles, we use Wikipedia as an ontology and obtain all the categories and general categories for the top n articles using a toolkit developed by [10]. For instance, a value "someone like you" has top Wikipedia categories "Adele (singer) songs" and "Singles certified septuple platinum by the Australian Recording Industry Association". These categories help to create the hidden topical structure that will be inferred using the observed attribute-values. Intuitively, the distribution of

[2]http://www.lemurproject.org/
[3]http://lemurproject.org/clueweb09/

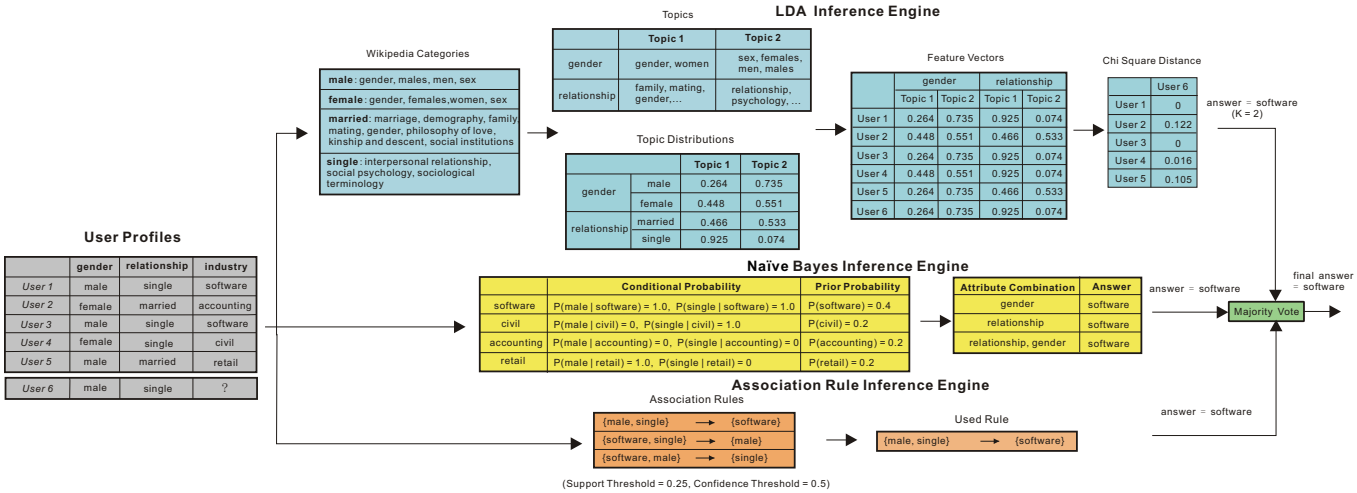

Fig. 2: A toy example where the adversary is attempting to predict the industry attribute-value for User 6 using public site data. The table on the left shows the available data. The remainder of the figure shows example scenarios for the LDA *(top)*, Naïve Bayes *(middle)*, Apriori *(top)*, and ensemble *(right)* inference engines.

attribute-values of a user to the topic space represents the likelihood that the user is interested in specific topics. With this probabilistic model, each user profile can be turned into a vector representation of topics and probabilities for each topic. We thus compare the similarity among users by calculating χ^2 values between these vectors.

Users that have a "similar" distribution over the topics can be identified by the adversary to infer missing attribute-values for a different attribute A_p. Suppose for user i, $\text{restricted}(i, A_p) = \top$, then the adversary can infer $\mathcal{P}_p^i$ by identifying k neighbors with the highest similarity to i in terms of the topic distribution for attribute A_q. Then, using a standard majority vote of the k neighbors for attribute A_p, a value for $\mathcal{P}_p^i$ can be inferred. We will discuss options for handling ties and alternative voting schemes in the next section.

Returning to our example, the top of Figure 2 shows the construction of the LDA-based Inference Engine. Our algorithm begins by determining the Wikipedia categories that map to the domains of the gender and the relationship attributes (see upper left table in Figure 2). Topics are extracted using these Wikipedia categories and a distribution for the attribute-values and the topics is determined (see 'Topics' and 'Topic Distribution' tables in Figure 2). Finally, after mapping the attribute-value topic feature vectors to the users for each attribute-value (see the 'Feature Vectors' table in Figure 2), the similarity between User 6's feature vectors and the other users is determined. Let's suppose $k = 2$. Since User 1 and User 3 have the smallest χ^2 distance to User 6, we assign their value for the industry attribute to User 6. In this case: software.

Naïve Bayes Inference. To contrast the LDA classifier, we also consider the Naïve Bayes [11] classifier that assumes an underlying independent attribute model. Using D' for every attribute A_j, this method begins by determining the probability of occurrence for all values in $|A_j|$. Suppose for user i, $\text{restricted}(i, A_j) = \top$. The adversary can infer $\mathcal{P}_p^i$ using the Naïve Bayes algorithm to make different predictions of $\mathcal{P}_p^i$ using different combinations of attributes in $D' - A_p$. A majority vote is taken over all the predictions to determine the final value for the hidden attribute, A_p.

Looking at our running example, Figure 2 shows the construction of the Naïve Bayes Inference Engine. The different probabilities are computed (see yellow colored tables in Figure 2). Computing all the predictions for the different attribute-value combinations and taking a majority vote over those predictions leads to assigning the value "software" to User 6 for the industry attribute.

Multi-Attribute Association Rules Inference. A new approach that we introduce in this paper involves generating multi-attribute association rules—i.e., rulesets containing attribute-values from one or more attributes in the data set. Each record $r = \langle id^k, v_1^k, v_2^k, \ldots, v_q^k \rangle$ in the subpopulation D' consists of the attribute-value pairs for some subset of attributes in D. We consider each such record a multi-attribute transaction, and compute the *frequent itemsets* in the subpopulation, where a frequent itemset is a set of attribute-value pairs in D' that occurs above a user specified minimum support threshold. A high level of support in D' is an indication that a particular itemset occurs regularly in the subpopulation. We then compute the confidences on the frequent itemsets, keeping those rulesets that are above a minimum confidence threshold, to produce the final set of association rules. For simplicity, we construct only single-inference association rules in which the antecedent derives exactly one consequent attribute-value. We remark that our association rule inference technique is agnostic to the particular frequent itemset algorithm.

We denote the set of association rules as $\mathcal{R}$. Each rule $R_k \in \mathcal{R}$ is of the form $(a_1, v_1), \ldots, (a_q, v_q) \Rightarrow (a, v, c)$, where q is a positive integer, $(a_1, v_1), \ldots, (a_q, v_q)$ and (a, v) are attribute-value pairs, and $c \in (0, 1]$ is the *confidence* of rule R_k. That is, the above rule can be interpreted as "the existence of attribute-values $v_1, \ldots, v_q$ in an itemset implies the existence of value v for attribute a with confidence c".

The adversary can infer the attribute-values using these

TABLE I: Example attribute properties on the subpopulation samples from LinkedIn ("L") and Google+ ("G").

Site	Attribute	Unique Values	Entropy (bits)	% Users w/ non-null vals
L	Headline	57716	14.28	100.0
L	Company Name	50124	15.29	75.14
L	Title	32020	14.07	75.22
L	Last Name	13010	10.85	100.0
L	Skill	10399	11.39	7.03
L	First Name	8224	10.34	97.18
L	Interests	4818	12.23	6.7
L	Location	1562	7.89	100.0
L	# Connections	501	6.40	100.0
L	Industry	147	6.41	87.32
L	Start Date (Year)	67	3.80	45.89
L	Start Date (Month)	12	3.53	41.46
G	Organization Name	54698	15.08	37.74
G	Display Name	34925	14.28	100.0
G	Places Lived	29469	13.97	34.69
G	Family Name	20387	12.38	96.34
G	Title at Organization	25262	13.88	29.85
G	Tagline	12031	13.54	17.32
G	Given Name	11387	10.28	96.34
G	Birthday	296	8.28	0.64
G	Relationship Status	9	2.09	7.37
G	Gender	5	0.97	86.59
G	Organization Type	2	1.00	37.74

rules. The adversary can choose to consider only the consequent of the rule having the highest support and confidence. Or he/she can consider all the rules that have at least a minimum support and confidence and take a majority vote. An instance of the Multi-Attribute Association Rules Inference is depicted at the bottom of Figure 2. Here, the frequent itemset algorithm outputs three rules that meet the minimum support and confident thresholds. The adversary applies the rule $\{\mathsf{male}, \mathsf{single}\} \rightarrow \{\mathsf{software}\}$ to a user who publicly reveals that he is male and single to infer that he is in the software industry.

Ensemble Inference. Because these different methods are complimentary in their approaches, we also consider an inference engine that combines the results of the LDA-based inference, the Naïve Bayes inference, and the Multi-Attribute Association Rules inference outputs. This ensemble approach uses the output of each of these different methods and determines the final prediction based on both the majority value and the confidence of the value.

V. Evaluation

We consider three metrics for evaluating the inference engine algorithms: inference accuracy, guessability, and inference gain. For a given attribute, we define *inference accuracy* to be the percentage of inferred values that correctly describe the targeted user. To quantify how often we are able to make inferences—regardless of whether or not they are correct—we define the *guessability* of an attribute as the fraction of public profiles for which we are able to guess the attribute's value. Finally, to assess the efficacy of the different methods, we introduce *inference gain*. Inference gain is the ratio of inference accuracy using our inference engine to inference accuracy achieved by guessing the most frequent attribute-value for a given attribute. Hence, inference gain provides intuition as to how well population-based inferences perform compared with the strategy of guessing the most popular value.

A. Subpopulation Sampling and Attribute Properties

We collected 88,085 public profiles from Google+ (a popular social networking site) and 91,150 public profiles from LinkedIn (a site designed for professional networking)[4]. Since neither site offers random subsampling features, we collected profiles by querying each service for random names. To obtain our list of random names, we searched for random profiles on Twitter—a service that uses sequential identifiers and hence permits straightforward random sampling. For each random name, we searched both Google+ and LinkedIn for matching public profiles using the services' APIs. LinkedIn and Google+ respectively returned up to 100 and 1,000 matching profiles for each query. All returned profiles were added to our corpus of sampled public profiles.

Because we need to both build and test our inference engine, we divide the Google+ and LinkedIn profiles into (1) our inference engine subpopulations (the training set), and (2) the site profiles of users that our adversary is attempting to reconstruct (the test set). Using the training set, we construct the prediction models and the association rules, while the test set data are used to measure the inference accuracy, guessability, and inference gain achieved by site-level inference. We use 90% of the data for D' and the remaining 10% for the test set.

To better understand the attributes that comprise the LinkedIn and Google+ subpopulations, we present the number of unique attribute-values, the Shannon entropy over the distribution of these values, and the percentage of users with non-null values in Table I. We observe that the size of the domains of the different attributes ranges from two to over fifty thousand and that while many values are disclosed by over 75% of our subpopulation (e.g., first name, last name, industry, gender and title), others are rarely disclosed (e.g., birthday, specialty, and skills). This variability is also reflected in the entropy of the attribute-values: attributes such as company name had significant uncertainty, while others had little variance (e.g., gender had less than a single bit of entropy).

Table I also provides insight into the type of information that users publicly disclose. Less than 10% of Google+ users publish their relationship status, indicating a trend towards keeping personal information hidden from the public. The propensity to disclose certain attributes also appears to be correlated to an online network's *specialty*. For instance, on Google+ — a site primarily dedicated to social networking — users generally felt comfortable revealing their gender (87%), but rarely disclosed their job title (30%) or their organization name (38%) to the public. In contrast, on the more career-oriented LinkedIn network, 75% and 87% of users respectively specified their title and industry.

B. Inference Engine Construction

We compare the performance the three inference algorithms as well as an ensemble of these methods. Prior to conducting the analysis, we remove attributes that are used internally by the sites and have little outside value (e.g., objecttype and profileurl). For our test, we assume that the complete set of

[4]To ensure compliance with the services' terms of service, we used the sites' official APIs, obeyed rate limits, and collected data using a single machine over a period of months.

attribute-value pairs for each user i represents user i's profile $\mathcal{P}^i$. We apply leave-one-out cross validation for the evaluation. Particularly, for each profile, we remove one attribute-value pair $(a_i,\ v_i)$ (i.e., we set $P^i = \mathcal{P}^i \setminus (a_i, v_i)$) and measure the inference engine's ability to infer this *hidden attribute*. We repeat this test for each attribute-value pair in the record.

C. Inference Accuracy

We begin our analysis by considering the accuracy associated with each method individually when using different parameters. This is followed by a detailed comparison of the different inference approaches. For this analysis our inference engine makes a prediction about a single consequent attribute.

LDA-based Inference. For the LDA-based inference model, we empirically found that basing the inference on one, two, or three attributes leads to the best accuracy results on these data sets. Therefore, for each consequent attribute, we generate a final result by taking the majority vote over these combinations of non-null attributes. When determining the final similarity between the targeted user and the other users in D', we use χ^2 similarity. There are two additional parameters to consider when employing the LDA-based approach: (1) which topics to use from Wikipedia—i.e., sibling level topics (self categories) or parent level topics (parent categories)—and (2) the number of nearest neighbors (k).

Figure 3 compares different approaches for generating topics based on the Wikipedia ontology: using article self categories, using article general (or parent) categories, or using both. The x-axis shows the hidden attribute being inferred and the y-axis shows the averaged inference accuracy for inferring the hidden attribute using the other attributes in the inference engine. Surprisingly, we find that for these data sets the inference accuracy is stable for the different topic models. We surmise that this is due to the narrow vocabulary semantics associated with our attributes Therefore, for the remaining experiments involving LDA, we use only self categories from Wikipedia since it is less costly to build. Finally, Figure 3 also shows that the inference accuracy for social attributes on Google+ are high, while the inference accuracy for professional attributes are high on LinkedIn.

Figure 4 depicts the inference accuracy for different attributes as k varies. The y-axis shows the inference accuracy as a function of the value of k (x-axis). If different neighbors give different answers, a majority vote is used to resolve conflicts[5]; if there is a tie, then no result is produced using this method. We observe that the accuracies generally improve by two to 10 percent, depending upon the attribute. All the attributes are fairly stable at $k = 5$. Therefore, for the remaining experiments involving LDA, we set $k = 5$.[6]

Naïve Bayes Inference. Our Naïve Bayes classifier makes inferences using (i) a single attribute, (ii) two attributes, or (iii) three attributes. A majority vote is taken over all the predictions to determine the final value for the hidden attribute. To understand how the strength of the majority vote influenced the inference accuracy, we compute a *confidence* score for each inference, where confidence is defined as the ratio between the most probable result returned by the majority vote and the second most probable result. For example, let's suppose we are interested in determining a user's favorite color. If our algorithm returns the following predictions: $probability(red) = 0.7$, $probability(blue) = 0.2$, then the confidence will be $0.7/0.2$.

Figure 5 shows the inference accuracy results of Naïve Bayes for different minimum confidence settings. When points are missing for an attribute, the inference accuracy at the particular confidence is 0. For Google+, when the minimum confidence rises from 1 to 2, the inference accuracy also improves significantly. This is also the case for the company industry and the industry attributes in LinkedIn. Once the minimum confidence reaches 7, the inference accuracy for all the attributes is relatively stable. Therefore, for the remaining experiments, we use a confidence of 7.

Association Rule-based Inference. We construct single-inference association rules using the Apriori [1] method that derive exactly one attribute-value pair. Association rules have two parameters: support and confidence.

Figure 6 shows the inference accuracy for Google+ and LinkedIn attributes when we vary the *support* threshold (x-axis) and the minimum confidence is fixed at 0.5. For Google+, the inference accuracy increases and then decreases as the support threshold increases. For LinkedIn, the inference accuracy is relatively stable across different support thresholds. Similarly, the inference accuracy for different *confidence* thresholds is also shown in Figure 6. For both Google+ and LinkedIn, inference accuracy increases with confidence, with the slope of the increase decreasing once the minimum confidence reaches 0.5. Both name related attributes on Google+ and industry related attributes on LinkedIn have an improvement of over 40% as the confidence increases.

Based on these empirical results, we use a minimum support level of 0.0025 and confidence of 0.5 for the remaining experiments. For this setting, our Apriori inference engine produced rulesets $\mathcal{R}_{G+}$ and $\mathcal{R}_{LI}$, respectively containing 1633 and 163 association rules for Google+ and LinkedIn. If we increase the minimum support and confidence levels beyond that, the number of rules generated decreases substantially. For example, for over half of the attributes, no inferences were possible when the confidence threshold was 70% or greater.

D. Inference Gain Comparison of All Methods

To study the performance of our inference techniques in more detail, we compare their performance to the simpler strategy of predicting the most frequent value for a given attribute. Figures 7 *(left)* and 8 *(left)* plot the inference gain of the four inference engines: LDA-based, Naïve Bayes, Association Rule, and Ensemble. (Recall that inference gain is the ratio of the inference accuracy achieved by the inference technique to the inference accuracy achieved by guessing the most popular value.)

For the Google+ attributes, we obtain inference gains of over 30 for more than half of the attributes. This is attributable

[5]The other voting strategy we considered was reciprocal rank, but its accuracy was lower than majority vote.

[6]As a comparison, we also considered a simpler term frequency model (building a TF-IDF index) using Wikipedia categories and then computed the cosine similarity of these Wikipedia categories. Due to space limitations, we do not show the results. However, we found that the LDA-based inference slightly outperformed the TF-TDF approach in terms of accuracy.

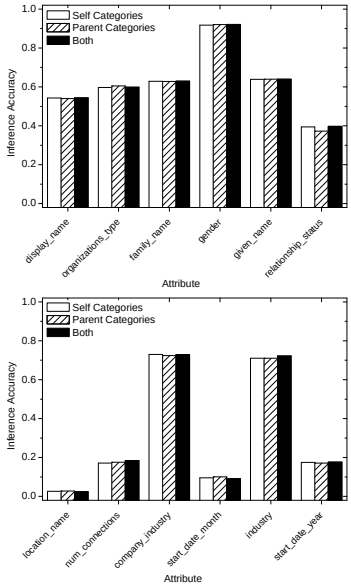

Fig. 3: Inference accuracy of using different Wikipedia topic categories with Google+ *(top)* and LinkedIn *(bottom)*.

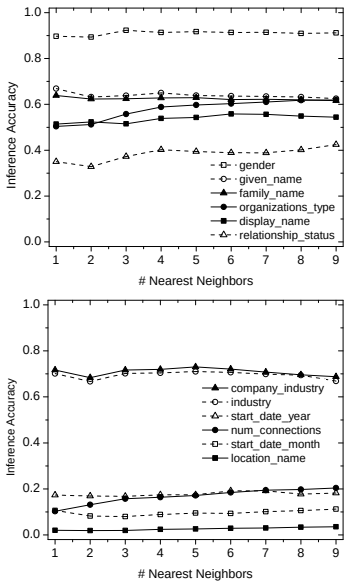

Fig. 4: Inference accuracy of LDA when varying k (number of nearest neighbors) for Google+ *(top)* and LinkedIn *(bottom)*.

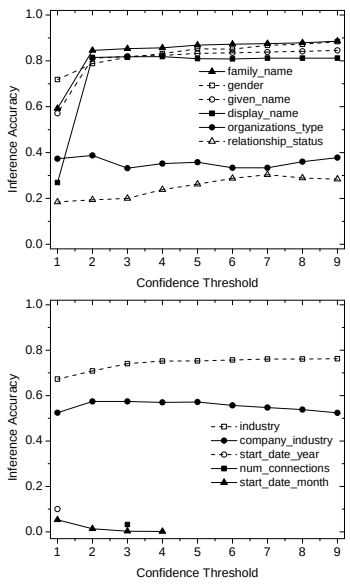

Fig. 5: Inference accuracy of Naïve Bayes as a function of confidence for Google+ *(top)* and LinkedIn *(bottom)*.

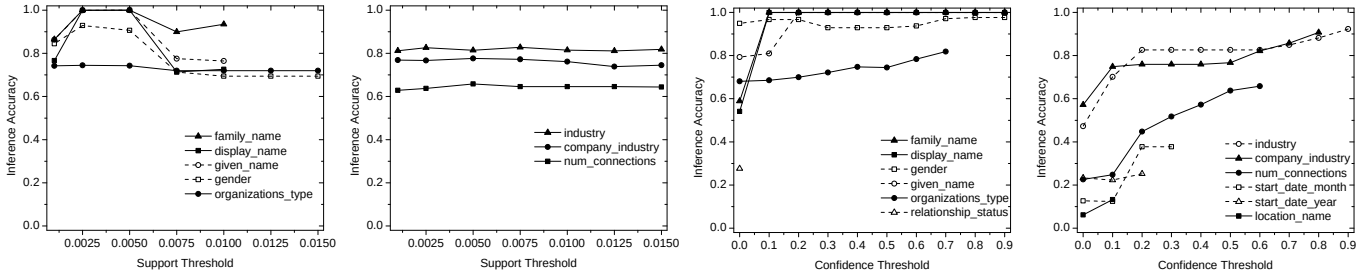

Fig. 6: Inference accuracy of association-rule based inference as a function of the support threshold and confidence threshold for Google+ *(first and third)* and LinkedIn *(second and fourth)*.

to the strong correlation between the three related name fields (i.e., family, given, and displayname). LinkedIn has a similar strong connection between the industry and company industry fields, but the inference gain is lower than that achieved with Google+ due to the (relatively) higher accuracy of guessing the most common attribute on those fields. The results also show that all of the tested inference techniques significantly outperform the naïve strategy of guessing the most popular value, highlighting the effectiveness of inference techniques for this problem. A site's privacy controls are *insufficient* to protect privacy given an adversary who can use the user's exposed attribute-values as well as a site's sampled subpopulation to accurately predict many unknown attributes.

E. Guessability

Figures 7 *(center)* and 8 *(center)* show, for each attribute on Google+ and LinkedIn, the fraction of public profiles for which the inference engine is able to make a prediction (i.e., its guessability). For both data sets, association rule mining has the lowest guessability. We attribute this to our selection of the minimum support, which led in many instances to there being no relevant association rules. In our parameter selection, we opt for correctness (i.e., inference accuracy) over guessability, and hence achieve the former at the expense of the latter.

In contrast, the LDA and ensemble methods provide the greatest guessability. When the adversary *needs* to construct a guess in a principled manner (regardless of the correctness of that guess), these methods are most appropriate. (A trivial method of achieving perfect guessability is to guess randomly for each hidden attribute; however, as is indicated by Figures 7 and 8 *(left)*, such a strategy achieves poor inference accuracy.)

In practice, the adversary likely desires both high inference accuracy and guessability. As a coarse measure of the overall utility of an inference technique, we consider the product of its guessability and inference accuracy. Figures 7 *(right)* and 8 *(right)* plot this product for the different attributes for Google+ and LinkedIn. The figures further highlight the advantage of LDA and the ensemble method: in nearly all cases, LDA and the Ensemble outperformed Apriori and Naïve Bayes.

VI. Conclusion

This paper investigates the degree to which site-based population data can be leveraged to correctly infer the *undisclosed*

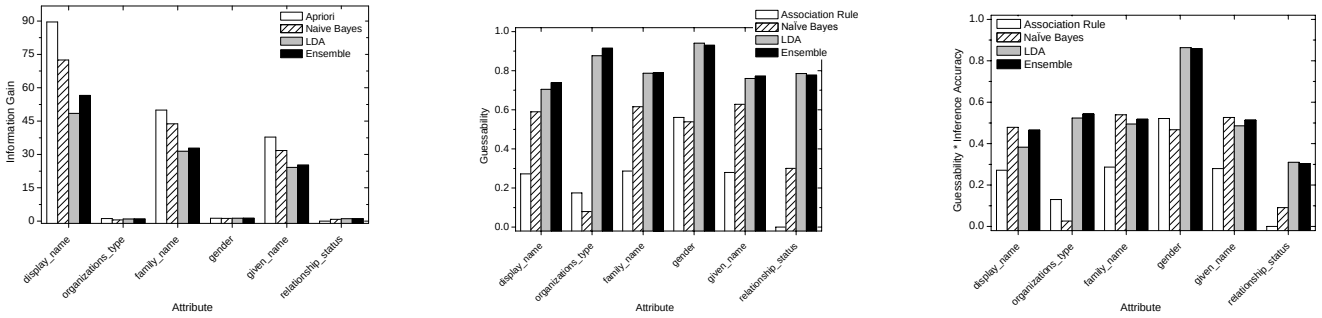

Fig. 7: Inference gain *(left)*, guessability *(center)*, and guessability times inference accuracy *(right)* for Google+.

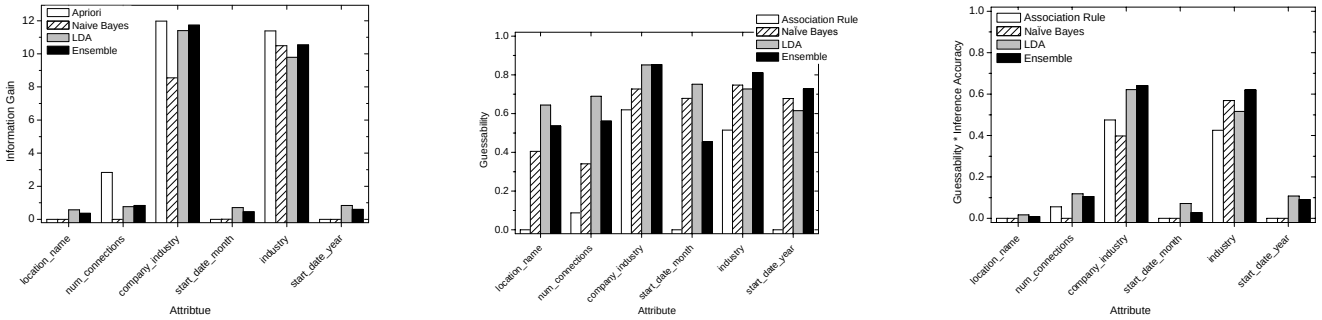

Fig. 8: Inference gain *(left)*, guessability *(center)*, and guessability times inference accuracy *(right)* for LinkedIn.

attributes of online users. Our methods leverage a targeted user's publicly disclosed attributes, as well as the patterns of relationships between publicly disclosed attributes on others on the site, to predict additional attributes of a targeted user. Similar to previous work, we demonstrate that different inference engines are able to predict withheld attributes of a user's profile with considerable accuracy. Our work differs from the previous work in this arena in terms of (1) the applied methodology, (2) the inference algorithms used, and (3) the data sets considered. This work is an important initial step toward understanding random site-based inference of personally hidden attributes. We also analyze the distribution of attribute-values across two large, real-world data sets. Examining the attribute-value distributions gives us additional insight into the types of data people feel comfortable publishing. In particular, our examination of the disclosed attributes and values on social networking (Google+) and professional networking (LinkedIn) sites highlights the semantic differences perceived by users of these services.

Acknowledgements: This work was supported in part by the National Science Foundation through grants CNS-1223825, CNS-1149832, CNS-1204347, and CNS-1064986.

REFERENCES

[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *International Conference on Very Large Data Bases (VLDB)*, 1994.

[2] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An Online Social Network with User-defined Privacy. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2009.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[4] A. Chaabane, G. Acs, and M. Kaafar. You Are What You Like! Information Leakage Through Users' Interests. In *Network and Distributed System Security Symposium (NDSS)*, 2012.

[5] D. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidence. In *Proceedings of the National Academy of Sciences (PNAS)*, 2012.

[6] R. Dey, Z. Jelveh, and K. W. Ross. Facebook users have become much more private: A large-scale study. In *IEEE PerCom Workshops*, pages 346–352, 2012.

[7] Diaspora. Diaspora: The Community-run, Distributed Social-network. https://joindiaspora.com/.

[8] J. Ferro, L. Singh, and M. Sherr. Identifying Individual Vulnerability Based on Public Data. In *International Conference on Privacy, Security and Trust (PST)*, 2013.

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

[10] D. Guan and H. Yang. Increasing stability of result organization for session search. In *European Conference on Information Retrieval (ECIR)*, 2013.

[11] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[12] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. In *Proceedings of the National Academy of Sciences (PNAS)*, 2013.

[13] A. Lenhart and M. Madden. *Teens, Privacy & Online Social Networks: How Teens Manage their Online Identities and Personal Information in the Age of MySpace*. Pew Internet & American Life Project Washington, 2007.

[14] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring Private Information Using Social Network Data. In *International Conference on World Wide Web (WWW)*, 2009.

[15] K. Liu and E. Terzi. A Framework for Computing the Privacy Scores of Users in Online Social Networks. In *IEEE International Conference on Data Mining (ICDM)*, 2009.

[16] S. Livingstone. Taking Risky Opportunities in Youthful Content Creation: Teenagers' Use of Social Networking Sites for Intimacy, Privacy and Self-Expression. *New media & society*, 10(3):393–411, 2008.

[17] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *ACM international conference on Web search and data mining (WSDM)*, 2010.

[18] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[19] A. Ramachandran, L. Singh, E. Porter, and F. Nagle. Exploring Re-identification Risks in Public Domains. In *International Conference on Privacy, Security and Trust (PST)*, 2012.

[20] D. Rosenblum. What Anyone Can Know: The Privacy Risks of Social Networking Sites. *Security & Privacy, IEEE*, 5(3):40–49, 2007.

[21] 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 *International Conference on World Wide Web (WWW)*, 2009.