Leveraging Network Structures to Reveal Obfuscated and Hidden Attributes in Google+ Networks[1]

I. Introduction

As social networks are increasingly used by companies, advertisers, and possibly malicious third-party actors, to make inferences about our social interactions, there exists a necessity for a strong understanding of privacy. However, there is a tension between sharing all information necessary for important analysis tasks, while successfully hiding all information that reveals a user's **protected attributes**. These protected attributes may include legally protected classes, such as race and gender, which could be misused to make biased and discriminatory decisions by companies, advertisers, or malicious actors such as election influencers. These attributes may also include sensitive information such as location, which can compromise an individual's personal safety and security.

Although past works on data de-anonymization focus on user re-identification, we believe that for many attack motives, simply de-anonymizing personal attributes is sufficient. For example, consider a social media platform that provides an advertising company information about the user's preferences and network, but hides protected attributes. If we can use structural features and preference attributes to infer the user's gender or location, then this attribute's privacy in compromised in ways the user did not allow. This has serious ramifications in many issues, such as online stalking - if publicly available network information, combined with the network's structural properties, is sufficient to predict location, that can create serious issues for individuals who seek to protect their location. Furthermore, for a small enough group, de-anonymizing personal attributes can result in re-identification. Thus, we do not focus on re-identification in this project, but instead focus on recovering and assessing privacy with regards to personal attributes such as **gender** and **location.**

The most standard anonymization technique found in social network datasets is by re-mapping attribute values to arbitrary tokens. In a classic social network, where nodes represent users and edges represent relationships, this can be viewed as replacing "Race: Asian" with "Race: race4" across all users. A stronger technique is also replacing *the attribute name itself*, so that "Race: Asian" would be replaced with "Attribute1: a1value4" across all users. Our aim is to examine the robustness of these obfuscation strategies by utilizing *1. network structure* and 2. *auxiliary unprotected information*. Auxiliary sources such as the U.S. Census Data can provide spatial distribution information showing how population characteristics correlate with geographical location. For example, we would expect features such as race, income level, and education level to be correlated with location. We therefore hypothesize that by leveraging these distributions, we can discover cross-correlation relationships with distribution statistics from the social networks themselves, possibly deanonymizing and inferring sensitive attributes.

A stronger censorship mechanism used to protect sensitive attributes is to entirely remove them, instead of replacing with arbitrary tokens as discussed above. In this case the recipient learns nothing about the distribution of the attribute across the population. However, since the data still reveals the structure of the social connections and the unprotected non-sensitive attributes, it may again be possible to make use of auxiliary information and non-censored exemplars from same or similar social network to learn models that could then be used to predict the deleted attributes.

Motivated by the preceding observations, in this project we explore how network structural features and unprotected attributes can predict sensitive attributes that have been protected either by obfuscation of values (i.e. semantically understandable values replaced one-to-one by arbitrary tokens) or complete deletion of the attribute from the published dataset. We specifically focus on two sensitive

[1] Associated code and data is available at https://github.com/meghabyte/cs224w_project

attributes: participant's **gender**, and characteristics associated with a participant's **location**. For location we specifically focus on three characteristics: which part of the US is the location in, what is the income level (i.e. low, middle, or high income), and the type of location (e.g. small town, mid-size city, large city etc.).

I.A. Problem Definition

Formally, the work in this proposal explores the two problems relating to learning values of sensitive attributes that have been protected by one of the two forms of anonymization mechanisms discussed above.

Problem 1: Attribute Deobfuscation Problem

We frame our first problem as an Attribute Deobfuscation Problem, where we aim to re-identify the values of some sensitive type of attributes that have been obfuscated for all the nodes. Concretely, we can consider a typical obfuscation scheme for **gender**, which is a common protected attribute, as follows:

replacing original value $gendervalue_{original} \in \{'male', 'female', 'other'\}$

with $gendervalue_{obfuscated} = f_{gender,\ obfuscation}(gendervalue_{original}) \in \{0,\ 1,\ 2\}$

where $f_{gender,\ obfuscation}$ is bijective and kept secret.

Note that the set $S = \{0,\ 1,\ 2\}$ of obfuscated values may be replaced with any other set of the same cardinality. Our objective is to identify $f_{gender,\ obfuscation}$ given the public dataset about the social network graph, and to similarly identify the functions obfuscating other protected attributes. The decision about which attributes to obfuscate is governed in part by laws (e.g. HIPAA, FERPA etc.) and in part by factors such as terms of use of the data provider and privacy sensitivity of the data publisher - therefore, our problem shows a failure of these laws in properly protecting user privacy. As many Social Networks datasets from resources like the Stanford Large Network Dataset Collection (https://snap.stanford.edu/data/), such as the Facebook datasets, are modified via obfuscation, we hope to show that these commonly used datasets use poor obfuscation techniques.

Problem 2: Hidden Attribute Inference Problem

In our second problem we aim to re-identify the values of a sensitive attribute which has been completely deleted prior to publication, or equivalently obfuscated by replacing all values with a single constant arbitrary value such as "unknown". Our objective is to show that by learning models over other social network data from similar population where the sensitive attribute was not deleted, we can make inferences about the deleted attribute. We hope to use correlations that exist between the deleted sensitive attribute and social connections, and other attributes that were deemed non-sensitive and thus published, or can be reasonably inferred with the help of auxiliary information. Specifically, we focus on **location**, and given a sanitized dataset about a social network from which location related information has been deleted, we aim to infer characteristics of the locations associated with a node in the dataset. Since locations come in very fine granularity (there are more than 29,000 named "places" in the US that are recognized by the US Census Bureau), we instead focus on inferring coarser-grained characteristics of the ground-truth location: median income level, population size, and geographic region or division in the US (as defined by the Census Bureau). We also assume that we have access to a separate non-sanitized data set from the social network on which we can employ machine learning algorithms to train models.

II. Related Work

Our approach to de-anonymization is motivated by the following two findings, which results in our approach being different from most previous works that focus on re-identification:

1. Many works have shown that information necessary for different inference tasks can be correlated with protected attributes that we seek to anonymize, such as street image information correlating with race, income, and political beliefs [Gebru17]. Therefore, those ensuring privacy must be aware of how such auxiliary information can break anonymization.
2. Similarly, past research has shown variance in network structure properties based on protected demographics. For example, [Barnes-Mauthe13] used network density and homophily detection to demonstrate the ability of clustering Hawaiian fisherman by race, while [Psylla17] found that males demonstrate less network homophily in social networks than females.

It is clear from a privacy angle, therefore, that social network data owners must be careful about how much information their dataset and network structure provides. There exists a large amount of previous work showing that ensuring the privacy of social network data is an important yet non-trivial task, with most approaches mainly focusing on re-identification.

For example, a canonical work in this area by Narayanan & Shmatikov [Narayanan09] analyzes anonymity in social networks by categorizing attacks, such as the attacker's resources, what auxiliary information is available, and whether the attack is active or passive. They argue that large-scale re-identification is facilitated by memberships in multiple social networks and ready availability of auxiliary information. One generic re-identification method they presented took advantage of two social networks, Twitter and Flickr, to predict nodes belonging to the same user and therefore achieve deanonymization due to the combined information. The method requires availability of a few seed mappings of users in the anonymous social network to this non-anonymous one, and exploits common structural properties between the two networks. They also discuss defense mechanisms such as using randomized tokens rather than actual identifiers of attributes, scrambling user profiles, publishing only a subset of attributes, injecting random noise into the graph structure, and methods such as k-anonymity. Similarly, [Srivatsa12] also make use of a social network as auxiliary information, but for the different purpose of deanonymizing mobility traces which they transform to an anonymized social contact graph with a time dimension as well.

Relative to these works, we focus instead on the Attribute Deobfuscation Problem - which seeks to re-identify values of obfuscated attributes in anonymous social networks by using publicly available auxiliary information about the attributes to model their relationship with the network structure, and across different attributes. We therefore do not rely on multiple social networks with the same individuals, but demonstrate a more general approach that can still result in large privacy violations.

There exists some previous research on using purely network structure for Deobfuscation, such as [Tang09], who use latent social dimensions to learn multiple diverse relations that are associated with the same network. [Gong11] presents methods that simultaneously use both network structure and node attribute information to inform tasks such as of prediction of links and node inference of missing and incomplete node attributes, and show that link prediction accuracy is improved by first inferring missing attributes. They introduce an attribute-augmented social network called Social-Attribute Network (SAN) which consists of social nodes and attribute nodes, with links existing between two social nodes (social links) or a social node and an attribute node (positive or negative attribute links to indicate presence or absence of a trait such as male gender,), or between two attributes to indicate mutually exclusive traits (e.g. male gender and female gender). Both the Relation Learning problem of [Tang09] and Attribute Inference Problem of [Gong11] have connections to our Attribute Deobfuscation Problem, but are different. Our problem can be viewed as one where instead of some attributes missing values while the rest have public values, we have all attributes being given values that are substituted with obfuscated identifiers, as described earlier.

Finally, [Liao18] presents a social network embedding framework which learns low-dimensional vector space representations for social actors by preserving both the structural proximity and attribute proximity, with the former capturing global network structure and the latter accounting for homophily effect. The authors report significant performance gain on tasks such as node classification compared to embeddings that do not explicitly incorporate homophily effects [Perozzi16]. In a recent work [Lee17] researchers from Princeton have presented more powerful structure-based de-anonymization attack methods for re-identifying individuals which, unlike current methods, do not require prior seed knowledge or suffer from the imprecision of such seed information. They use multi-hop neighborhood information and enhanced machine learning techniques. While their focus is on re-identifying individuals in an anonymized graph by making use of a non-anonymized auxiliary graph, their methods are helpful for our Attribute Deobfuscation Problem as well.

III. Approach

III.A. Identification and Preparation of Data

The first step of our approach is identification and preparation of data. In order to evaluate the success of our overall approach, it is important that we have access to the ground truth, which presented the conundrum that publicly available datasets about social networks already obfuscate protected attributes. Moreover, many of the unprotected attributes are in free form unstructured text that would require extensive preprocessing to be useful. After considerable investigation, we decided to use a dataset drawn from the Google+ Social Network. This dataset, found at http://snap.stanford.edu/data/ego-Gplus.htm, contains information about ego-networks from the Google+ platform. The dataset contains 107,614 nodes with 13,673,453 edges across ego-nets for 132 users, across the years 2011-2012. The edges represent friend relationships on Google+, and each node consists of the following attributes in a one-hot vector encoding (# of unique values in parentheses): *gender (3), institution (2426), job_title (5886), last_name (2623), place (5612), and university (2494).* The dataset file contains 30,494,866 edges, but many are repeated as there are actually 13,673,453 unique edges. We therefore modify the file with sort | uniq before processing.

We also identified a dataset from http://home.engineering.iastate.edu/~neilgong/gplus.html that covers an entire Google+ network of users, rather than simply providing egonets. This dataset contains four Google+ Snapshots from 2011, two before and two after Google+ was open to public. Each snapshot has both directed social structure and node attributes. The first snapshot has 4,693,129 nodes and 47,130,325 edges, which represent social links. Each node has the following attributes: *Employer, School, Major, Places Lived.* As gender is not included, this dataset can only be used for our second problem, namely inferring characteristics of location. We were only able to make minimal use of this data set because of its sheer large size with > 4M nodes made it too compute intensive to work with.

Establishing ground truth values for the protected **location** and **gender** attributes required some pre-processing. We performed the following steps:

1. *Take advantage of weak obfuscation to label ground-truth gender for the SNAP Google+ data.* While gender values were “anonymized”, we took advantage of an obfuscation flaw by the dataset creators and the fact that we knew whether two nodes shared the same gender to create a straightforward manual labeling process. This is described concretely in the *Methods* section.
2. *Clean the location labels in the Google+ data so that they can easily be mapped to Census and Income data.* This is initially tricky because the locations are provided in completely free text. We used a combination of the following Python Packages that automated city/state/country tags for approximately 90% of the dataset (we manually labeled the few hundred remaining):

- Geotext https://geotext.readthedocs.io/en/latest/
- US Metadata https://pypi.org/project/us/
- Pycountry https://pypi.org/project/pycountry/

3. Match all Google+ Location data with Auxiliary Information about locations (income level, population, etc.) from U.S. Census Data, as described below.

Auxiliary information about locations

While previous research relied on auxiliary information in the form of other social networks, one of our main contributions is taking advantage of US Census Bureau information. As this information is always publicly available and was not released specifically in the context of academic research in Computing/Information sciences, it is especially powerful as a form of rich yet common auxiliary information.

We obtained auxiliary geographical information from the US Census Bureau's *American Fact Finder* website [AFF18] which aggregates data from censuses to provide information about the United States, Puerto Rico and the Island Areas. Many geographical entities of different granularities are available, e.g. all of US, by state, by place, by zip code, by census tract etc. In particular, we downloaded information from tables B01003 (total population) and B19013 (median income) for various granularities (states, regions, places).

Since the SNAP Google+ data is from 2011-2012 timeframe, we used the 5-year average tables from 2012. We wrote python code to read and normalize all the data, and to provide convenient functions so that given a place string as entered in the Google+ dataset, we can easily fetch various characteristics of that place, such as median income, population, and geographical region. Further, we mapped median income and population to coarser bins, with median income being binned into bottom 30%, middle 40%, and top 30%, and the population being binned into village, town, large town, city, large city, and metropolis by following the settlement hierarchy convention used by landscape historians [Wikipedia18].

III.B. Algorithms and Methods

We developed methods corresponding to the two problems identified earlier in this report: 1. the Attribute Deobfuscation Problem, and 2. the Hidden Attribute Inference Problem.

The Attribute Deobfuscation Problem has an important distinction from node classification or attribute inference. In the latter, the objective is to make an independent inference about the true value of an attribute for a node given the social structure as well as typically the value of that attribute for other nodes (which allows network homophily to be exploited). In contrast, in attribute deobfuscation we have additional knowledge about subsets of nodes which have the same (though unknown) value for the obfuscated attribute, and the objective is not to make independent inferences but to discover the one-to-one mapping used by the unknown obfuscation function while making use of information not only for the public dataset but also any auxiliary information from external data sources. In essence, this is an additional constraint that needs to be satisfied.

Likewise, we also note that the Hidden Attribute Inference Problem too is different from the standard node classification problem because none of the nodes in the provided subgraph, which has been sanitized via protected attribute deletion, are annotated with values for the protected attribute. As such some the techniques presented in the course (Lecture 16, which addressed the problem: *"Given a network with labels on some nodes, how do we assign labels to all other nodes in the network?"*) are not applicable since in the absence of any label one cannot initiate the collective inference process. Instead one has to rely solely on inference models learnt elsewhere to make an initial inference about a node's label, and then additionally exploit homophily to refine it as part of a network level collective inference as described in the class. Using these insights, we developed the following three methods relating the Attribute Deobfuscation, and the Hidden Attribute Inference problems.

Method 1: Exploiting correlation between obfuscated and non-obfuscated attributes

As noted earlier, in many datasets a subset of attribute types is treated as sensitive and thus subject to obfuscation. This leads to the possibility that publicly available auxiliary information may provide information about the joint distribution of an obfuscated attribute to various non-obfuscated attributes. While preprocessing the SNAP Google+ ego-network dataset, we encountered an opportunity to exploit this approach to deobfuscate gender. Specifically, the dataset has 3-valued obfuscated gender for a subset of nodes, as well as multivalued *unobfuscated* lastnames for a subset of nodes. The last names are presumably unobfuscated under the reasoning that they cannot be used to re-identify an individual. However, many users mistakenly enter their first names as last names, and *these are correlated with gender*. Making use of lists of first names from https://www.scrapmaker.com, we made an initial prediction of the gender from the last names provided.

We treat each such successful prediction as a vote towards whether the obfuscated gender value for that node corresponded to ‘male’ or ‘female’, while inability to make a prediction was considered a no-vote. Failure of prediction could happen for three reasons: 1. last name was not provided, 2. last name was ambiguous and appeared in the list of both male and female first names, and 3. last name did not appear on either of the two lists. Lastly, we looked for large clusters in the voting data in order to make an assignment of obfuscated gender values to actual gender values. For example, if we see two large and dominant clusters where obfuscated value ‘0’ got a large number of votes from nodes with last names occurring in the male names list but not in the female names list, while the obfuscated value ‘1’ got a large number of votes from nodes with last names from the female names list but not in the male names list, then we would predict with very high confidence that ‘0’ deobfuscated to ‘male’ and ‘1’ to ‘female’. This approach leads to the following algorithm, results from which on the selected dataset are reported in Section IV.

```
def DeobfuscateGender1($D$ : dataset with gender obfuscated to values 0 or 1):
        $M$ = set of male first names
        $F$ = set of female first names
        $VoteCount = \{1 : \{‘male’ : 0, ‘female’ : 0\}, 2 : \{‘male’ : 0, ‘female’ : 0\}$
        for every node $v$ in the D:
          if $v.lastname \in M \wedge v.lastname \notin F$ :
            $VoteCount[v.gender]['male']++$
          elif
            $v.lastname \in F \wedge v.lastname \notin N$ :
            $VoteCount[v.gender]['female']++$
        if $VoteCount[0]['male']$ and $VoteCount[1]['female']$ are dominant:
          return $\{1 : 'male', 2 : 'female'\}$
        elif $VoteCount[1]['male']$ and $VoteCount[0]['female']$ are dominant:
          return $\{2 : 'male', 1 : 'female'\}$
        else:
          return FAILED
```

Method 2: Modeling relationship between social structure and an obfuscated attribute

The network homophily theory suggests that correlations between node attributes are reflected in social network structure. For example, nodes with very large in-degrees may provide estimates of gender and location, as might the structure of a node’s neighborhood. Indeed, works such as [Tang09], [Gong11], and [Liao18] cited earlier make use of this relationship between social structure and attributes. A simple initial approach can be to define a feature vector for a node capturing attributes of its relevant neighborhood, and then learn a model to predict the obfuscated attribute using the feature vector.

We computed feature vectors which contained the following values (these were obtained via feedback from experiments), which were selected to focus on ratios and thus be independent of absolute size as we intend to apply this method to gender, where the ratios are more important than absolute value:

- *in degree / total degree of ego node* $v_i = \frac{|\{e_{ji}:v_j \in V, e_{ji} \in E\}|}{|\{e_{ji}:v_j \in V, e_{ji} \in E\}|+|\{e_{ij}:v_j \in V, e_{ij} \in E\}|}$
- *out degree / total degree of ego node* $v_i = \frac{|\{e_{ij}:v_j \in V, e_{ji} \in E\}|}{|\{e_{ji}:v_j \in V, e_{ji} \in E\}|+|\{e_{ij}:v_j \in V, e_{ij} \in E\}|}$
- *edges internal to egonet of* v_i */ total edges with a node in egonet of* $v_i = \frac{|\{e_{jk}:v_j, v_k \in N_i, e_{jk} \in E\}|}{|\{e_{jk}:v_j \in N_i \vee v_k \in N_i, e_{jk} \in E\}|}$
- *edges going out of egonet of* v_i */ total edges with a node in egonet of* $v_i = 1 - \frac{|\{e_{jk}:v_j, v_k \in N_i, e_{jk} \in E\}|}{|\{e_{jk}:v_j \in N_i \vee v_k \in N_i, e_{jk} \in E\}|}$
- *local cluster coefficient (directed) of ego node* $v_i = C_i = \frac{|\{e_{jk}:v_j, v_k \in N_i,\ e_{jk} \in E\}|}{k_i(k_i - 1)}$
    - Where k_i is the number of vertices, $|N_i|$ in the neighborhood of vertex v_i
- *fraction of neighboring nodes of ego node* v_i *with same obfuscated value* $= \frac{|\{v_j:v_j \in N_i, gender(v_j) == gender(v_i)\}|}{|N_i|}$

However, a model learnt over these features will yield independent predictions for the various test samples of node with obfuscated attributes. Therefore, we developed a simple algorithm where we treat each prediction as a vote, and then for each group of samples with the same obfuscated value of the attribute we compute the winning prediction along with a confidence metric that is the ratio of the winning vote count divided by the total number of votes. As the last step we use the winning vote and confidence metric to assign a unique unobfuscated label to each obfuscated value, using the confidence metric to break ties. This approach leads to the following algorithm, experimental results from which on the selected dataset are reported in Section IV.

```
def DeobfuscateGender2(
  $D_{train}$ : dataset with true gender,
  $D$: dataset with gender obfuscated to values 0 or 1
):
    $clf$ = classifier trained over $D_{train}$ using features described above
    $g_p$ = gender for each node in $D$ predicted by $clf$ ($g_P \in \{'male', 'female'\}$)
    $g_O$ = obfuscated gender for each node in $D$ ($g_O \in \{0, 1\}$)
        $VoteCount = \{1 : \{‘male’ : 0,\ ‘female’ : 0\},\ 2 : \{‘male’ : 0,\ ‘female’ : 0\}$
        for every node $v$ in the D:
            $VoteCount[g_O[v]][g_P[v]]$ ++
        $b[1]['male'] = \frac{VoteCount[1]['male']}{VoteCount[1]['male'] + VoteCount[1]['female']}$
        $b[1['female'] = \frac{VoteCount[1]['female']}{VoteCount[1]['male'] + VoteCount[1]['female']}$
        $b[2]['male'] = \frac{VoteCount[2]['male']}{VoteCount[2]['male'] + VoteCount[2]['female']}$
        $b[2['female'] = \frac{VoteCount[2]['female']}{VoteCount[2]['male'] + VoteCount[2]['female']}$
        $x, y = argmax_{i,g} b[i][g]$
        if $(x == 1 \wedge y == 'male') \vee (x == 2 \wedge y == 'female')$:
            return $\{1 : 'male',\ 2 : 'female'\}$
        else:
            return $\{1 : 'female',\ 2 : 'male'\}$
```

Method 3: Modeling relationship between social structure and a hidden attribute

Our last method focused on the problem of inferring values of hidden attributes, focusing in particular on the problem of making inferences about various characteristics of the places that the user may have been located in. Our data set provided ground truth in the form of the locations that a user lives in (currently or in the past). We derived (as described earlier in Section III.A) characteristics of these locations using data from US Census Bureau’s *American Fact Finder* website. We then used these as hidden attributes, and explored how well we could predict them using features derived from structure and obfuscated gender. We targeted the following place characteristics:

- *place_income_category*: Median income, categorized as bottom 30%, middle 40%, top 30%)
- *place_type*: Type of place, categorized into seven (village, town, large town, city, large city, and metropolis) by following the population based settlement hierarchy convention used by landscape historians [Wikipedia18].
- *place_region_id*: Region as defined by US Census Bureau (Northeast, South, Midwest, West, and Island Territories)
- *place_division_id*: Region as defined by US Census Bureau (New England, Middle Atlantic, East North Central, West North Central, South Atlantic, East South Central, West South Central, Mountain, Pacific, and Island Territories)

We use the following features, with the main difference relative to Method 2 (which targeted gender) being that we also made use of absolute values of followers and followed nodes since characteristics of a location are likely to effect that as well. The features were as follows:

- *in neighbors of ego node* $v_i = |\{e_{ji} : v_j \in V, e_{ji} \in E \wedge e_{ij} \notin E\}|$
- *out neighbors of ego node* $v_i = |\{e_{ij} : v_j \in V, e_{ij} \in E \wedge e_{ji} \notin E\}|$
- *inout neighbors of ego node* $v_i = |\{e_{ij} : v_j \in V, e_{ij} \in E \wedge e_{ji} \in E\}|$
- *fraction of in neighbors of ego node* $v_i = |\{e_{ji} : v_j \in V, e_{ji} \in E \wedge e_{ij} \notin E\}| / |\{e_{ji} : v_j \in V, e_{ji} \in E \vee e_{ij} \in E\}|$
- *fraction of out neighbors of ego node* $v_i = |\{e_{ji} : v_j \in V, e_{ij} \in E \wedge e_{ji} \notin E\}| / |\{e_{ji} : v_j \in V, e_{ji} \in E \vee e_{ij} \in E\}|$
- *fraction of inout neighbors of ego node* $v_i = |\{e_{ij} : v_j \in V, e_{ij} \in E \wedge e_{ji} \in E\}| / |\{e_{ji} : v_j \in V, e_{ji} \in E \vee e_{ij} \in E\}|$
- *local cluster coefficient* (*directed*) *of ego node* $v_i = C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$
  - Where k_i is the number of vertices, $|N_i|$ in the neighborhood of vertex v_i
- *gender* = *obfuscated gender attributed* (use an extra value to indicate private gender)
- *neighborhood gender profile*: vector of length 9 indicating how many of the three types of neighbors (in, out, inout) have each of the three possible obfuscated gender values (1, 2, 3)

Using these features we trained a classifier (specifically, a decision tree classifier) to predict the desired place related characteristic. The results from experiments using this algorithm on the selected dataset are reported in Section IV.

IV. Results and Findings

IV.A. Deobfuscating Gender using DeobfuscateGender1 (Method 1, Section III.B)

The results of applying the DeobfuscateGender1 method to our dataset are shown in the table below. In the dataset, the obfuscated gender attribute had three values, and the DeobfuscateGender1 was accordingly adapted. We know that Google+ allows for a custom 'other' gender as well besides 'male' and 'female', along with an option for keeping the gender hidden. While in the absence of ground truth we cannot be sure, there is extremely strong evidence from the table below that the obfuscation function mapped 'male' to 1, 'female' to 2, and 'other' to 3. Moreover, with this deobfuscation, we are able to then infer gender for another 27,372 + 11,345 + 955 = 39,672 social network nodes whose genders were kept private. Overall, we correctly predicted gender for 78,073 out of 107,614, or 72.5% of the individuals, allowing us to deobfuscate gender for all individuals.

	Prediction from Last Name = Male	Prediction from Last Name = Female	Last Name Ambiguous or Private
Obfuscated Gender = 1	**28,219**	165	27,372
Obfuscated Gender = 2	81	**7,951**	11,345
Obfuscated Gender = 3	233	105	**955**
Gender Hidden	1,147	510	29,541

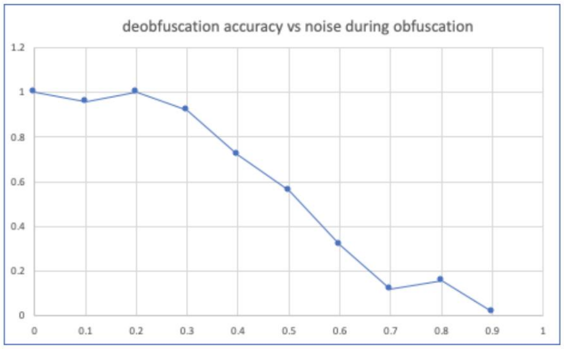
To ensure that the assignment of ‘1’=’male’ and ‘2’=’female’ is consistent, we manually examined the 165 cases where our strategy predicted ‘female’ but obfuscated gender was 1, and the 81 cases where our strategy predicted ‘male’ but gender was 2. In most of these cases the explanation was simply that the names were valid last names as well, e.g. ‘Grant’, particularly when taking into account non-Western names. Still there are cases that we cannot explain, such as a user with last name ‘Jake’ with obfuscated gender ‘2’, and our hypothesis is that either these are unusual last names or that the user changed their gender or last name. However, we emphasize that we have < 1% such cases. We did not examine cases corresponding to obfuscated gender being 3 because in those cases the names did not correlate with gender.

The dataset offers other opportunities as well, since the combination of last names with place and university or institution can provide additional deobfuscation opportunities with the help of open-source information on the web. As cleansing information such as university or institution requires a lot of manual effort, we could not explore this, but in principle our ideas apply there as well. Moreover, the method can be used to make some inferences about attributes that are not even present in the database, such as age, race, education etc. This suggests that in light of the readily available auxiliary information about relationships across attributes, successful obfuscation requires that all attributes in the database be obfuscated since even a single attribute type, howsoever innocuous, that is left unobfuscated can potentially lead to inferences about private information.

IV.B. Deobfuscating Gender using DeobfuscateGender2 (Method 2, Section III.B)

Method 2 also targets gender but seeks to deobfuscate without relying on the leakage of gender information through the last names. The Google+ dataset we use is organized in terms of egonets of 132 nodes out of a total of 107,614 nodes. While the overall graph is weakly connected and has a large strongly connected component, the 132 ego nodes have different characteristics than the other nodes. We focused the initial exploration only on the ego nodes, and used the gender inferred using Method 1 as the ground truth, which was possible only for 117 nodes. Using the features, a Decision Tree Classifier, and the voting approach as described earlier under Method 2, we conducted 50 trials with 5-fold cross validation. Across these trials the mean node level gender inference accuracy (considering the gender provided by Method 1 as ground truth) was 95.8%. As the data set was considerably skewed (5x more males than females), we also conducted experiments where we trained over a training set obtained with a modified sampling strategy to have a balanced training set, and obtained similar accuracies.

However, as discussed earlier, our task is not independent inference of genders but rather coming up with a consistent assignment of ‘male’ and ‘female’ labels to the two obfuscated values ‘1’ and ‘2’. Across 50 trials, we obtained 100% success with our voting strategy that is part of Method 2, and across 1000 trials our success rate was 99.5%. In effect, the network level constraint provided by obfuscation helps correct for errors in node level inferences. To assess the importance of leveraging homophily, we also evaluated the accuracy while excluding $fraction\ of\ neighboring\ nodes\ of\ ego\ node\ v_i\ with\ same\ obfuscated\ value$, from our feature vector which is the only feature that exploits homophily. Without it, the accuracy (with voting to incorporate network level constraint) drops to less than 70%, thus suggesting that homophily has a significant impact on accuracy.



Impact of noisy obfuscation defense

Clearly, straightforward obfuscation where true attribute values are simply mapped one-to-one to semantically opaque labels can be easily deobfuscated, either by making use of unprotected attributes that are correlated with genders (Method 1) or by models that exploit correlation with structural properties and homophily. As a defense mechanism one can consider adding noise to the obfuscation process whereby gender values are probabilistically altered before being obfuscated. The noise will affect the decision made by the voting mechanism which tries to satisfy network level constraint. We evaluated the impact of noise (probability of flipping gender when obfuscating) on correctness of deobfuscation, and obtained the adjacent plot which shows that the deobfuscation process is resilient upto gender flip probability of 30%, and to make deobfuscation accuracy no better than random guess one needs to flip with 50% probability, making the obfuscated data useless since no meaningful distribution can be obtained.

IV.C. Inferring Characteristics of Hidden Location Attribute (Method 3, Section III.B)

The final experiment focused on applying Method 3 to infer characteristics of the locations that the user might have lived in, using only social connection and gender information. Here we faced the additional challenge that only about 50 out of 132 ego nodes had one or more locations that was in the US, with others either not providing location information or having locations in non-US location about which we unfortunately did not have a good source of auxiliary information similar to the US Census Bureau. Our investigation is therefore based on this smaller set of users, and while the results are promising, establishing their statistical validity requires additional experimentation. Based on 20 trials with 5-fold cross validation we obtained the inference accuracies in the table below for the various location related characteristics. Note that the ground truth for some of the users listed multiple places that the user had lived at. For such users, during training, we created replica samples corresponding to different labels, and during testing we considered the prediction to be correct if it matched the corresponding characteristic of any one of the places the user had lived at.

Location Property	place_division_id	place_region_id	place_income_category	place_type
Accuracy	0.67	0.69	0.95	0.81

V. Conclusions

In this project we explored methods to infer node attributes that may have been obfuscated or suppressed in a published dataset. We used a combination of network structure, homophily, correlation between obfuscated and public attributes that is enabled by auxiliary information from public sources, and models based on exploiting auxiliary information about geographical distribution of population characteristics from census data. Our methods perform extremely well in the case of obfuscated attributes where shared obfuscated values across nodes provide additional constraints that help correct for errors in node level inferences, and are quite resilient to addition of even significant amounts of noise during obfuscation. This indicates that obfuscation via remapping semantically understandable values one-to-one to arbitrary strings is a poor strategy if the the goal is to provide privacy as it can easily be broken, and to make it work requires addition of a large amount of noise in order to render the obfuscated data useless.

In the specific case of our dataset, a defense against the first attack would be to check that users are not swapping their first and last names, or alternatively replacing last names by hash. Our results on the inference of hidden or suppressed attributes (characteristics of locations that the user might have lived at) are promising but less trustworthy on account of the limited data that we had, and warrant further investigation. Moreover, the model used by our method only makes use of social structure and distribution of public or obfuscated attributes within the local one-hop neighborhood. Were appropriately large data sets to be available, a potential direction of future investigation would be to learn models of how the target attributes are distributed across the network (e.g. along random walks) and use this information as constraints to correct the independent node level inferences.

VI. Code on GitHub

The code for the project, in the form of two iPython notebooks - one to preprocess data and the other to perform analysis - is available at https://github.com/meghabyte/cs224w_project.

VII. References

[AFF18] US Census Bureau, "American FactFinder," https://factfinder.census.gov/ (accessed November 12, 2018).

[Bailey18] Bailey, Michael, Rachel Cao, Theresa Kuchler, Johannes Stroebel, and Arlene Wong. "Social Connectedness: Measurement, Determinants, and Effects." J. of Economic Perspectives 32, no. 3 (2018): 259-80.

[Barnes-Mauthe13] Barnes-Mauthe, Michele, Shawn Arita, Stewart D. Allen, Steven A. Gray, and PingSun Leung. "The influence of ethnic diversity on social network structure in a common-pool resource system: implications for collaborative management." Ecology and Society 18, no. 1 (2013).

[Gebru17] Gebru, Timnit, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, et. al. "Using deep learning and google street view to estimate the demographic makeup of the us." arXiv preprint arXiv:1702.06683 (2017).

[Gong11] Gong, Neil Zhenqiang, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, et. al. "Jointly predicting links and inferring attributes using a social-attribute network (san)." arXiv preprint arXiv:1112.3265 (2011).

[Koppel07] Koppel, Moshe, Jonathan Schler, and Elisheva Bonchek-Dokow. "Measuring differentiability: Unmasking pseudonymous authors." Journal of Machine Learning Research 8, no. Jun (2007): 1261-1276.

[Lee17] Lee, Wei-Han, Changchang Liu, Shouling Ji, Prateek Mittal, and Ruby B. Lee. "Blind de-anonymization attacks using social networks." In Proc. of the Workshop on Privacy in the Electronic Society, pp. 1-4. ACM, 2017.

[Liao18] Liao, Lizi, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. "Attributed social network embedding." IEEE Transactions on Knowledge and Data Engineering (2018).

[Narayanan09] Narayanan, Arvind, and Vitaly Shmatikov. "De-anonymizing social networks." In Security and Privacy, 2009 30th IEEE Symposium on, pp. 173-187. IEEE, 2009.

[Perozzi16] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855-864. ACM, 2016.

[Psylla17] Psylla, Ioanna, Piotr Sapiezynski, Enys Mones, and Sune Lehmann. "The role of gender in social network organization." PloS one 12, no. 12 (2017): e0189873.

[Srivatsa12] Srivatsa, Mudhakar, and Mike Hicks. "Deanonymizing mobility traces: Using social network as a side-channel." In Proceedings of the 2012 ACM Conf. on Computer and communications security, pp. 628-637, 2012.

[Tang09] Tang, Lei, and Huan Liu. "Relational learning via latent social dimensions." In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 817-826. ACM, 2009.

[Wikipedia18] Wikipedia contributors, "Settlement hierarchy," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Settlement_hierarchy&oldid=864868516 (accessed November 12, 2018).