On Exploiting Innocuous User Activity for Correlating Accounts Across Social Network Sites

Oana Goga[†], Howard Lei[*], Sree Hari Krishnan Parthasarathi[*], Gerald Friedland[*], Robin Sommer[*§], and Renata Teixeira[†]

TR-12-008

May 2012

Abstract

This paper studies whether it is possible to identify accounts on different social networks that belong to the same user just by using publicly available information in a user's posts. In particular, we explore three features to capture a user's online activity: the geo-location attached to a user's posts, the timestamp of posts, and the user's writing style as captured by language models. Our analysis, based on correlating user accounts across Yelp, Flickr, and Twitter, shows that such otherwise innocuous features can indeed enable attackers to track users across site boundaries. This result has significant privacy implications as users tend to rely on an implicit notion that social networks remain separate realms. Moreover, current privacy controls remain insufficient to contain the risk of cross-site correlation.

† CNRS and UPMC Sorbonne Université, 4 place Jussieu, 75005 Paris, France

* International Computer Science Institute, 1947 Center St., Ste. 600, Berkeley, California, 94704

§ Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, California, 94720

This work was partially funded by the National Science Foundation (NSF) through grant number CNS: 1065240 ("TC: Medium: Understanding and Managing the Impact of Global Inference on Online Privacy"). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. INTRODUCTION

Users of online social networks increasingly scrutinize privacy protections as they realize the risks that sharing personal content entails. Typically, however, much of the attention focuses on properties pertaining to *individual* sites, such as specific sharing settings Facebook offers or Google's terms of service. What users tend to miss, though, is a broader threat of attackers correlating personal information *across site boundaries*. While on a per-site basis, a user may deem fine what she posts to her Facebook, Twitter, and LinkedIn accounts, she might be revealing much more than she realizes when considering them in *aggregate*. As one example, a social engineering attack could first identify employees of a victim organization on LinkedIn, and then examine their Facebook accounts for personal background to exploit while also following their tweets to understand travel patterns. Indeed, we already see legitimate business models based on such correlation techniques, such as services offering "social media screening" to weed out job applicants (e.g., [1]).

In this work we set out to advance understanding of such correlation attacks. In general, it is much harder to defend against cross-site inference than to protect personal information on individual social networks where privacy settings directly control what becomes public. As combined data sets can often reveal non-obvious relationships—as prior work on deanonymization [2] convincingly demonstrates—it remains challenging to assess the correlation threat even for sophisticated users. More fundamentally, as a research community we lack insight into what precisely enables correlation attacks to succeed, along with counter-measures one can take for protection.

To further our understanding, we pursue a case study that examines the initial step of any correlation attack: *identifying* accounts on different social networks that belong to the same user. In contrast to past work, we focus on exploiting implicit features derived from a user's *activity*, rather than leveraging information explicitly provided—and hence more easily controlled—such as name or date of birth. Specifically, we explore matching accounts based on *where*, *when* and *what* a user is posting. As it turns out, combining these three types of features provides attackers with a powerful tool for targeting individuals.

As our example setting, we examine account correlation across Twitter, Flickr, and Yelp, for which we demonstrate that they provide sufficient public information for us to link user accounts. For our study, we deliberately choose social networks where account correlation is unlikely to cause much concern. However, similar techniques apply to more sensitive targets as well, in particular to sites where users expect to remain anonymous such as on dating sites, job portals, and special-interest forums.

Furthermore, we devise a possible set of attack heuristics, yet we emphasize that our choices are far from exhaustive. We also do not strive to fully automate our attacks, but rather take the perspective of an attacker targeting a specific individual. In that setting, identifying a small candidate set of accounts on other networks is sufficient to allow for manually sifting through for the correct match. Finally, extensive ground truth is hard to come by for commercial social networks, and we thus limit our evaluation to a reduced set of users for whom we can externally determine their account relationships.

We profile users with three implicit features of their activity: the geo-location attached to a user's posts; the timestamps of a user's posts; and the user's writing style modeled with a probabilistic approach. After discussing our methodology in §2, we first evaluate the potential of each of these three features individually to match a user's accounts across sites (in §3, §4, and §5, respectively). Then, we evaluate the improvements in accuracy that result from combining all three features (§6). Our results show that, when available, location and timestamps are powerful for correlating accounts across sites. Although we cannot necessarily find an exact match between a user's accounts on two different sites, we can often narrow down to a relatively small set of possible matches. For example, for 70% of users for whom we have ground truth we can match their Yelp account with a set of at most 1,000 possible accounts in Twitter, which is small compared to the total number of Twitter accounts. For 35% of the users, this set can even be reduced to 250 possible Twitter accounts.

A user's language model is not as effective by itself, but it helps further reduce the set of candidate matches when combined with other features. Generally, we find that our attacks work better for some users than for others, which we leverage for gaining insight into the properties that enable successful correlation and, hence, for giving recommendations on what users may do to undermine it.

2. METHODOLOGY

Our overall goal concerns understanding how user activity on one social network can implicitly reveal their identity on other sites. In §2.1, we first discuss *features* that we derive from user activity to build characteristic activity profiles. In §2.2 we then introduce our basic threat model: an attacker with moderate resources targeting a specific individual. We discuss the data sets we use for evaluation in §2.3, and metrics for measuring attack performance in §2.4.

2.1 Features

For our case study, we choose three types of features for building activity profiles that are present on many social networks: location, timestamps and language characteristics.

Location: Many social networks provide location information directly in the form of geotags attached to user content, potentially with high accuracy if generated by GPS-enabled devices like mobile phones. However, even without geotags, one can often derive locations implicitly from posted content (e.g., when users review a place on Yelp, that gives us an address). Furthermore, a number of online services map images and textual descriptions to locations or geographic regions (e.g., by identifying landmarks) [3, 4, 5, 6]. For our study, we use the *location profile* of a user, i.e., the list of all locations associated with her posts on a specific social network. The intuition behind that choice is that the combination of locations a user posts from may sufficiently fingerprint an individual across sites.

Timestamps: Many mobile services and applications such as `Foursquare`, `Gowalla`, and `Instagram` allow users to automatically send content to multiple sites simultaneously. The resulting posts then have almost identical timestamps, which we can exploit to link the corresponding accounts.

Language: The natural language community has demonstrated that users tend to have characteristic writing styles that identify them with high confidence [7]. While these methods typically work best with longer texts, such as blog

posting or articles, we also examine them with shorter inputs to understand if they can contribute to correlation attacks on sites such as Twitter.

2.2 Attacker Model

As our basic threat model, we assume a *targeted individual*: the attacker knows the identity of his victim on one social network, and she wants to find further accounts elsewhere that belong to the same individual. More precisely, for two social networks SN_1 and SN_2, we assume having one account $a \in SN_1$ and aiming to identify account $b \in SN_2$ so that $user(a) = user(b)$.

We assume that we are facing an attacker with moderate resources–e.g., with access to a small number of computers and the ability to rent further cloud services for a limited period of time. For such an attacker, it is not practical to compare the known account a with *all* accounts of SN_2 as that would require exhaustively crawling the target network. Hence, we assume an attack that proceeds in three steps. First, the attacker selects a *subset* of accounts $\widetilde{SN}_2 \subset SN_2$ that will plausibly include b. She then measures the similarity between a and all the $b_i \in \widetilde{SN}_2$ using an appropriate metric $(a, b) \in SN_1 \times \widetilde{SN}_2 \longrightarrow s(a, b) \in \mathbb{R}$. Finally, he selects the account $\hat{b} \in \widetilde{SN}_2$ that is most similar with a:

$$\hat{b} := \underset{b_i \in \widetilde{SN}_2}{\operatorname{argmax}} \quad s(a, b_i),$$

The attack is successful if $\hat{b}$ equals b.

Besides defining an appropriate metric (which we discuss in the following sections), a successful attack requires selecting a candidate set $\widetilde{SN}_2$ so that $b \in \widetilde{SN}_2$ while keeping its size sufficiently small to allow for collecting features from all of the included accounts. The key to that is selecting the accounts in $\widetilde{SN}_2$ based on the features considered. For example, if the attacker aims to link accounts by their location, she may assume that users who post regularly from within a certain region will most likely live there, and thus their postings on other sites will originate there as well. He can then build $\widetilde{SN}_2$ by extracting all users from SN_2 who have posted from that region. Likewise, if she strives to link accounts based on timestamps, he may select $\widetilde{SN}_2$ as those accounts for which she finds a temporal overlap with postings from $a \in SN_1$. Furthermore, one can also select accounts based on multiple features at once.

2.3 Data Sets

For our case study, we analyze correlation attacks with data collected from the three social networks Flickr, Twitter, and Yelp. We choose these sites because of their popularity and because they represent different types of social networks: photo sharing, micro-blogging, and service reviewing. We note that users will *not* necessarily consider account linking across these networks as a compromise of their privacy; however, a similar approach would apply to other, more sensitive sites as well. In the following, we describe the sets of users we select for our evaluation and the information we collect about them.

To assess the performance of our attacks, we collect a ground truth set of users for whom we know their accounts on the three sites. We obtain this set by exploiting the "Friend Finder" mechanism present on many social networking sites, including the three we examine. As the Finders often return pages that embed HTML in extensive Javascript, we use browser automation tools (Watir and Selenuim) to extract the results. We give the Friend Finders an existing list of 10 millions emails[1] and check if the emails correspond to accounts on any of the networks. Table 1 shows the number of *active*[2] accounts we have identified for each social network, as well as the corresponding number for the intersections. The second column, $GT_{Location}$, shows the number of accounts that have location information attached to at least one post, either as geotags or, for Yelp, in the form of addresses.

	GT_{Active}	$GT_{Location}$	$GT_{BayArea}$
Twitter	93,839	4,311	239
Flickr	59,476	4,826	379
Yelp	24,176	24,176	4,937
Twitter-Flickr	6,196	396	27
Twitter-Yelp	2,363	342	57
Flickr-Yelp	2,497	476	75
Twitter-Flickr-Yelp	569	70	8

Table 1: Number of users in the GT dataset.

Given the ground truth set, we could evaluate correlation attacks by directly following the attacker model discussed in §2.2: for each ground truth user, we collect corresponding sets $\widetilde{SN}_2$ from a target social network; and our attack would then identify an account $\hat{b} \in \widetilde{SN}_2$ as a likely match. However, this would require us to collect *separate* sets $\widetilde{SN}_2$ for each ground truth user, which is not feasible.

Instead, we limit our evaluation to users living in the San Francisco Bay Area, which allows us to use a *single* set $\widetilde{SN}_2$ for all of them. We identify this subset $GT_{BayArea} \subset GT_{Location}$ as those users who have more posts inside the San Francisco Bay Area than outside of it. We also limit the language and timestamp analyses to this subset. We note that such a geographical pre-filtering is consistent with what an attacker in the wild might do as well: inferring where a victim lives tends to be easy and hence location gives an obvious hint to reduce the size of the candidate set for language and timestamp.

We obtain the corresponding sets $SN_2^{BayArea}$ by crawling the three social networks for users from the Bay Area. We do not strive for completeness but instead emulate an attacker aiming to get a sufficient set of accounts from the targeted region that likely includes his victim. In this paper, our focus is to correlate Yelp and Flickr accounts to Twitter accounts. Thus, we only present how we obtain the $SN_2^{BayArea}$ set for Twitter.

Twitter provides a search API to get the posts around a specific longitude/latitude but it only returns results for a maximum of one week. This makes it unusable to get a satisfying set of users who live in the BayArea. Instead, we use the Streaming API[3] to collect in real-time all the tweets tagged with a Bay Area location. We collect all the tweets from the Bay Area during October 2011 to November 2011.

[1]This list comes from an earlier study by colleagues analyzing email spam. The local IRB approved collection and usage.

[2]Users often create accounts on social networks but never post anything; we only include accounts with at least one posting.

[3]While the Streaming API generally returns only a sample of tweets, limiting a query to a region the size of the Bay Area seems to indeed return the complete set.

We then use all 26,204 users who have sent at least one of these tweets. We find 75% of the Twitter $GT_{BayArea}$ users are already included in this set [4].

Finally, for all the $GT_{BayArea}$ and $SN_2^{BayArea}$ accounts, we download the publicly available profile information from the corresponding social network, including text, timestamps, and location of each posting. For Twitter, we use its API to get all the tweets and the metadata attached. Flickr's API likewise provides us the metadata attached to the photos. For Yelp, we again manually crawl and parse the profile pages.[5] The median number of tweets in $GT_{BayArea}$ and $SN_2^{BayArea}$ is 26 and 49 per user, respectively. For Yelp, we find 4 and 33 reviews per user, respectively. For Flickr, the medians are 10 ($GT_{BayArea}$) and 151 ($SN_2^{BayArea}$) photos. The total set of tweets, reviews, and images covers the time interval from 2007 and 2011.

2.4 Performance Metrics

We use two metrics to measure the performance of an attack. Recall from §2.2 that an attacker chooses $\hat{b}$ from all $b_i \in \widetilde{SN}_2$ so that it maximizes similarity with $a \in SN_1$. If successful, $\hat{b} = b$.

Our primary performance metric determines the number of similarity scores higher than $s(a, b)$, which we term a user's *rank* for a given attack:

$$rank(a, b) := \#\{b_i \in \widetilde{SN}_2 : s(a, b_i) \geq s(a, b)\} \qquad (1)$$

$rank(a, b) = 1$ means the matching is perfect and the attacker will pick the right account $\hat{b} = b$ directly. Since a perfect matching is however hard to obtain, we typically check if the *rank* is below a threshold X, i.e., the correct answer is part of the top X matches. For small X, an attacker can inspect that set manually.

The second metric is a verification metric, which considers false alarm rate and miss verification error given all pairs of similarity scores. Here, we consider the set of all similarity scores $s(a_i, b_i)$, where $a_i \in SN_1$ and $b_i \in \widetilde{SN}_2$. Some of these pairs correspond to user *matches*, i.e., $user(a_i) = user(b_i)$. After first establishing a threshold for $s(a_i, b_i)$, we consider matches below as *misses* and non-matches above as *false alarms*. Tuning for a low false alarm rate allows attackers to target a larger set of users simultaneously, with high probability that reported matches will be correct. Hence, we primarily consider the threshold at a 1% false alarm rate and then examine the miss rate. In addition, we also examine the *Equal Error Rate (EER)*, for which we establish the threshold such that miss rate and false alarm rate are equal. The EER is informative in terms of the general discriminability of our correlation approaches, but may not be too useful in narrowing down the range of user pairs that are matches.

2.5 Limitations

We emphasize that we see our work as an initial step towards a better understanding of cross-site correlation. Primarily, we aim to explore the *potential* and provide evidence that such attacks are feasible when relying on activity features that are hard to control for users. As such, we are less concerned about the specific tuning at which our heuristics yield their best results, nor do we claim that, e.g, thresholds we derive apply universally. Our set of ground truth is, in fact, too small to come to such conclusions. Rather, we explore the qualitative nature of our correlation techniques, and we make headway towards understanding the characteristics of the features we capture that facilitates their success.

Furthermore, we note that the approaches we use assume that we have an entire set of accounts for pairs of social networks, with which we can apply any technique or analysis to determine matches between user pairs. Due to the limited amount of data, our approaches do not assume the existence of an independent dataset for developing and tuning our techniques. The results we provide in this paper suggest certain best-case scenarios, and future implications for the power of our approaches when more data would likely become available. Nevertheless, our results are useful in identifying important directions that attackers can take in correlating social network accounts.

3. LOCATION PROFILES

We first examine location information in more detail. Our goal is to understand the degree to which locations attached to user content are sufficiently unique to identify an individual. Matching locations involves two parts, which we discuss in turn: *(i)* representing a user's location profile in the form of a fingerprint suitable for comparison; and *(ii)* defining a similarity measure between two such profiles. For evaluation, we focus on matching accounts from the Yelp and Flickr $GT_{BayArea}$ sets to the Twitter $SN_2^{BayArea}$. Based on the results, we also investigate what properties enable correlating users successfully by their location profiles.

3.1 Building Profiles

To motivate the use of locations, we start by examining the set of zip codes that user content originates from. Out of the 26,204 Twitter users in $SN_2^{BayArea}$, 23,395 exhibit sets of zip codes that are *unique*. Almost all the users without unique sets come with only a very small number of zip codes: 2,155 have only one zip code, 571 have two zip codes, 66 that have three, and 12 have four. For more than four zip codes, all the sets are unique except for 5 accounts that have more than 14 zip codes. We manually investigated these, and we found them to belong to 2 users both maintaining separate personas—which, incidentally, means we just linked related accounts by their location information. For Flickr, out of 1,907 users (we collected from the BayArea) only 66 don't have a unique set of zip codes. Out of them, 61 have only one zip code and 5 have two. For Yelp, out of 10,076, only 181 are not unique; 26 of them have one zip code, 120 have two, 31 have three, and 4 have four zip codes. Thus, given the large number of unique sets, we conclude that locations may indeed fingerprint users well.

Encouraged by that observation, we define a user's *location profile* as a histogram that records how often we observe each location in her posts. The histogram's bins represent "location units", such as zip code, city, or coordinates in an appropriate longitude/latitude grid. We normalize all histograms such that they represent probability distributions of posts in a particular "location unit". We also investigated other ways than histograms to represent location profiles but found them less effective. In particular, using lat-

[4]A set of users taken from the Bay Area from August 2010 to December 2011 achieves 95% coverage.

[5]Our collection is subject to the limits that the Twitter and Flickr API impose on the number of query results; 3200 and 1500, respectively.

itude/longitude directly and computing the Euclidean geographical distance is sensitive to the small deviations of geo-coordinates within a user's activity profile.

As location units, we test three different types of choices:

Administrative Region: We map each latitude/longitude geo-coordinate to an address using the Bing Maps API. Trying alternative address granularities (streets, zip codes, cities, counties, states), we find zip codes yielding the best results.

Grid: We map each latitude/longitude geo-coordinate to the cell within a spatial grid with the center closest to the coordinate. Considering cell sizes ranging from 1x1 km^2 to 12x12 km^2, 10x10 km^2 proves most effective in our experiments.

Clusters: As a more dynamic scheme to group geo-coordinates into regions, we use a clustering approach that considers population densities. A small cluster represents a popular small area (e.g., blocks of downtown San Francisco), while larger clusters represent bigger, less populous regions (e.g., a park or forest). Using the k-means algorithm with an Euclidean distance, we group latitude/longitude geo-coordinates from all users in $SN_2^{BayArea}$ into corresponding clusters. We then associate a specific location with the N closest clusters. We found that using N=20 produces the optimal results. We assign weights to each of the N clusters based on a Gaussian distribution.

Figures 1a and 1b compare the performance of different location representations at their best configurations. The plots show the cumulative distribution function (CDF) of the rank (see §2.4) for all pairs (a, b) corresponding to $GT_{BayArea}$ users, matching Yelp and Flickr with Twitter, respectively. The plots use a standard Cosine distance to measure the similarity between histograms (in the next section, we explore alternative choices). The solid line represents the minimum rank (the number of scores strictly greater than the GT score), the dashed line represents the maximum rank (the number of scores that are greater or equal to the GT score).

We see that the zip code profiles perform the best for Yelp to Twitter, and the longitude/latitude clusters perform the best for Flickr to Twitter. The grid-based profiles perform worse than zip codes, perhaps due to the fact that the latter better reflects population densities and places of interest than uniform grid cells. The cluster-based profiles perform worst for Yelp to Twitter matching and slightly better for Flickr to Twitter matching than the zip codes, but are far more expensive to compute. We believe, the cluster approach might get better results for countries and regions where zip codes do not reflect population density.

We also examine the verification EERs and miss rates at the 1% false alarm (FA) rate for the zip code and longitude/latitude clusters and grid approaches for Yelp to Twitter and Flickr to Twitter, shown in Table 2. The results confirm that the zip code and longitude/latitude clusters approach perform better in general. We note that the Flickr to Twitter correlations have lower FA rates than Yelp to Twitter correlations, implying that the former is easier.

We also examine different time intervals over which to build location profiles: one month, one year, two years, three years, and everything. Our results show that by aggregating at smaller time intervals we remove many data points from the profiles, such that the profiles become less precise.

Table 2: *EER and miss rate (at 1% false alarm) verification results for Yelp to Twitter and Flickr to Twitter correlations based on location profiles.*

Yelp to Twitter		
Method	EER(%)	Miss rate at 1% FA
Zip code	32.3	**70.2**
Long/Lat clusters	28.1	73.7
Long/Lat grid	28.1	89.5
Flickr to Twitter		
Method	EER(%)	Miss rate at 1% FA
Zip code	29.6	85.2
Long/Lat clusters	31.8	66.7
Long/Lat grid	29.6	85.2

While doing so helps to better identify a few prolific users, it impacts most others negatively.

Conclusion: Zip code-based profile representation provides the best trade-off between accuracy and computational cost. Building profiles over all of the available time range generally performs best and allow lower miss rates.

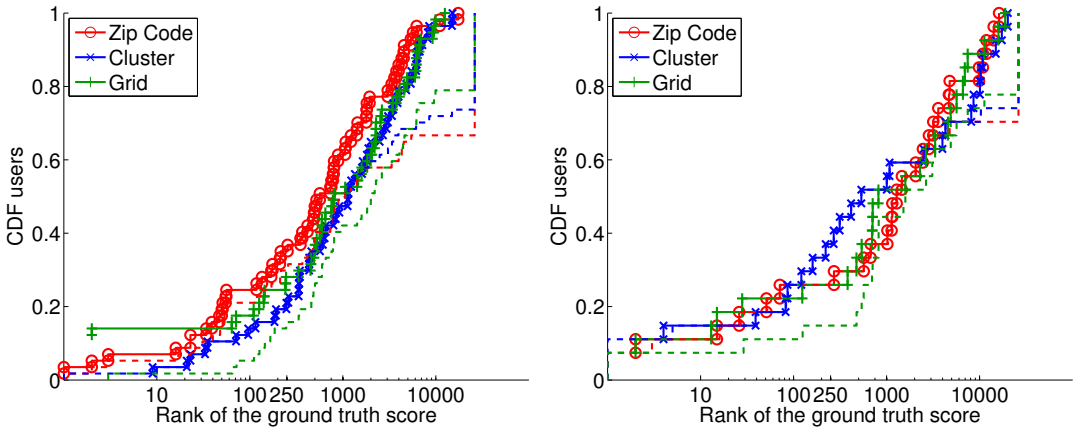
3.2 Similarity Metrics

So far we have used a Cosine distance to compare the histogram-based location profiles. We now investigate further choices to compare histograms of zip codes. The statistics literature offers a variety of metrics for measuring the similarity between two probability density functions P and Q [8]. We test a series of candidates, including Cosine and Jaccard from the Inner Product family; Euclidean and Manhattan from the Minkowski family; Hellinger from the Squared-chord family and Kullback-Leibler (KL) divergence from the Shannon Entropy family (see formulas in §A).

Our analysis shows that Cosine, Jaccard and Hellinger distances have similar performances, $\approx 20\%$ of users with ranking less than 50, 35% less than 250, and 70% less than 1000. The Euclidean distance has much higher rankings than other distances (75% of users have ranking above 1000). This is because the Euclidean distance is sensitive to the difference between the two values of a histogram's bin, and especially sensitive to large differences. In contrast, similarity metrics such as Cosine are sensitive to bins with non-zero values in both profiles, which better suites our goal. We use the Cosine distance for the rest of the experiments.

One issue with similarity metrics is that some accounts $b_i \in \widetilde{SN_2}$ have high scores even when they do not share many locations with the account $a \in SN_1$. This happens because all the metrics normalize their results by the total number of data points, which would give a higher score to b_i with fewer locations over another account, say $b_j \in \widetilde{SN_2}$, with more locations. This occurs even if b_j has more locations in common with a than b_i.

We address this issue by introducing two weights to penalize accounts with few common locations with account $a \in SN_1$. The mathematical definition of these weights are in Appendix B. The first weight, $weight_1$, considers if the set of common locations between a and $b_i \in \widetilde{SN_2}$ is popular in other accounts $b_j \in \widetilde{SN_2}, j \neq i$ and a. It gives lower weight if the set of common locations is popular, because it is harder to uniquely identifying a good match. The second weight, $weight_2$, accounts for the popularity of each loca-



(a) CDFs for Yelp to Twitter.

(b) CDFs for Flickr to Twitter.

Figure 1: CDF of the rank of the ground truth score for $GT_{BayArea}$ users for different location representations for Yelp to Twitter, and Flickr to Twitter. The solid and dashed lines represent the minimum and maximum ranks of the Twitter users for each approach.

tion within the sets of common locations across all account pairs (a, b_i). For example, if a and b_i have a very popular location in common such as Downtown San Francisco, $weight_2$ will give less weight to this location than if a and b_i shared a less popular location such as East Palo Alto. Applying $weight_1$ to the Cosine distance increases the number of users with ranking less than 50 by $\approx 7\%$ and applying $weight_2$ increases the number of users with ranking less than 250 with $\approx 10\%$. As we discuss in the next section, the main advantage of applying the weights is that the weighted metrics lead to more consistent predictions of the ranking from simple parameters.

Conclusion: The Cosine, Jaccard, and Hellinger distances perform best in our experiments. We modify these metrics with two weights to account for differences in popularity of locations.

3.3 Implications

The previous sections show that location profiles based on zip codes using cosine*weight as similarity metric perform the best to match accounts that belong to a single user in Yelp and Twitter, as well as in Flickr and Twitter. We focus on the zip code curves in Figures 1a and 1b to study the performance of correlating accounts with location alone. Say that an attacker is willing to search more exhaustively at a set of at most 1,000 candidate Twitter accounts. By searching the first 1,000 accounts based on the location ranking using the location profile of a user's Flickr account, he would find the correct Twitter account for 60% of our ground truth users; this percentage increases to 70% for Yelp accounts.

Our verification miss rates of 70.2% and 66.7% at 1% false alarms for Yelp to Twitter and Flickr to Twitter also have strong implications. Consider as a toy example a small company with 10 employees, each having a Flickr and Twitter account. There is a total of 100 Flickr vs Twitter account pairs, among which 10 correspond to matches between the users and 90 correspond to mis-matched users. Roughly one of those scores fall above the 1% false alarm scoring threshold. Given the 66.7% miss rate, roughly 7 out of the 10 scores of matching users fall *below* the threshold, while 3 fall above the threshold. Hence, if we only consider scores above the 1% false alarm threshold, 3 of them are matches, while 1 is a mismatch. This means that, in this simple scenario, 30% (3 out of 10) of the actual matches are detected and 75% (3 out of 4) of the retained pairs correspond to actual matches. Given the volume of information on the internet, there are likely many other scenarios in which our techniques can be used by a layman attacker targeting a set of users rather than a particular individual.

From the user's perspective, it is important to understand the properties of her location profiles which help or prevent the attacker from successfully correlating her accounts. Although the location profile is a powerful feature to correlate accounts of a single user across sites, some of our ground truth users in Figures 1a and 1b have high rank and so the attacker would have to search tens of thousands of accounts before finding the correct match. A user concerned about his privacy would like to fall in this category.

We now investigate the impact that an account's location properties have on the success of our matchings to develop a set of guidelines for users to avoid falling victim to such attacks. Take an individual user with two accounts $a \in SN_1$ and $b \in SN_2$. We study the following parameters that could impact an attacker's ability to match a and b: number of posts on each ($\#a$, $\#b$), the number of zip codes in their location profiles ($\#z_a$,$\#z_b$), the minimum number of posts on either account ($min(\#a, \#b)$), and the number of common zip codes between the location profiles ($\#z_{ab}$). To establish the importance of each of these parameters, we compute the Pearson correlation between the similarity scores and each of these parameters for Yelp ground truth users in the Bay Area (i.e., $GT_{BayArea}$) versus all Twitter users in the Bay Area (all users, not only ground truth). We find that $\#z_{ab}$ and $\#z_a$ (as well as $\#a$, which is strongly correlated with $\#z_a$) show the highest correlation. This result is intuitive given our definition of similarity metric incorporates the number of common locations directly, but it helps us understand possible defenses for users.

For that, we consider the probability that a user with $\#z_a$ zip codes in SN_1 and $\#z_{ab}$ common zip codes between SN_1

and SN_2 will be within the first M best matches:

$$P(\ rank(a,b) \leq M \mid \#z_a, \#z_{ab}\ ). \quad (2)$$

Figure 2 plots the probability that $rank(a,b) \leq threshold$ as a function of the number of zip codes shared between the two accounts (each line corresponds to a different threshold value: 50, 250, 500, and 1000). Figure 2a shows the probability for accounts with $\#z_a \leq 10$; Figure 2b shows $10 < \#z_a \leq 100$, and Figure 2c shows $\#z_a > 100$. We see that *(i)* as expected, the match quality increase with larger numbers of common zip codes; and *(ii)* an increase of $\#z_a$ inversely impacts the probability of a good match for a given number of common zip codes.

These results are interesting as they allow a consistent prediction of the matching performance from simple parameters of the accounts. For instance, for a pair with at most 10 review zip codes and 7 common zip codes, the ranking is less than 50 with probability 40% and the ranking is less than 500 with probability almost one (see Figure 2). This interesting feature is due to the weight that we added to the cosine metric to account for the different location popularity. Using the cosine metric alone does not give such consistent results. For instance, for a pair with at most 10 review zip codes and 7 common zip codes, the ranking is less than 50 with probability 3% and the ranking is less than 500 with probability 10%. This is because many accounts $b_i \in \widetilde{SN_2}$ can have a good similarity score if they share a few common locations with a and do not have many tweets. In contrast, using the weight will give low scores to these accounts. The trade-off is: with the cosine metric alone, some pairs with few frequent common locations will be well matched "by chance" but some pairs with rare common locations will be badly matched, whereas with the weight, pairs with frequent common locations may be missed but pairs with rare common locations will be well matched. As we have discussed in the previous section, the overall matching performance is fairly similar. However, the weight allows much more predictable matching performance.

Guidelines: From a user's perspective, these results suggest two strategies to avoid being vulnerable to matching. The first, obvious one is that it helps to avoid posting from the same zip code to separate social networks. The second, more interesting one suggests that one can correct past mistakes (i.e., already having many common zip code between accounts) by *adding* further locations to the first social network SN_1 (i.e., where one assumes the attacker to already know one's identity). Doing so effectively blurs the link to other networks by adding noise. A corollary is that posting from a series of different locations, like when on travel, remains unproblematic as long as one updates only one social network.

4. TIMING PROFILES

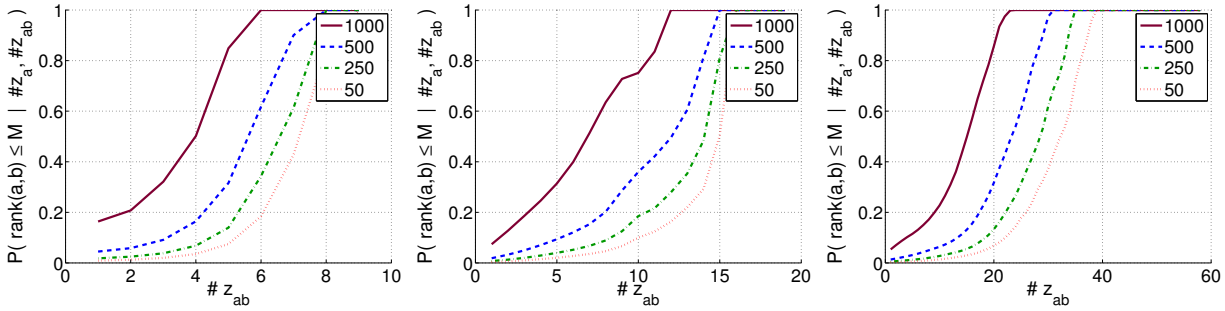
Many third-party applications, in particular on mobile devices, allow users to automatically send updates to different social networks simultaneously. For example, when Instagram uploads to Flickr, it can automatically tweet a pointer to the photo. We exploit this behavior to correlate the involved accounts, based on the timestamps of such automated postings.

In this section, we focus on our Flickr and Twitter data sets as Yelp does not provide sufficiently accurate timing information. Generally, we aim to find accounts where one or more timestamps of Flickr photos equals the timestamps of the tweets. However, even for simultaneous postings, timestamps may differ slightly due to processing delays and desynchronized clocks. Hence, we allow for a small window around two values when matching. The main question then is what an appropriate threshold is; a choice too small might miss actual post matches, while a larger threshold will report many matches corresponding in fact to unrelated posts.

To answer that question, we investigate the timestamp differences we see in our ground truth set, considering all the $GT_{BayArea}$ Twitter-Flickr pairs. For each pair (a,b), $a \in Twitter$, $b \in Flickr$ where $user(a) = user(b)$, we define the set of timestamp differences $td(a,b)$ as the set of differences between timestamps of two consecutive posts on different social networks. This set contains all the timestamp differences between posts on the two social networks potentially corresponding to the same content (e.g., a photo on Flickr and its link on Twitter). For example, assume that $tstmps(a) = \{t_1, t_2, t_3\}$, $tstmps(b) = \{T_1, T_2, T_3\}$ and the combined timeline is $\{t_1, t_2, T_1, t_3, T_2, T_3\}$; then the set of timestamp differences is $td(a,b) = \{T_1 - t_2, t_3 - T_1, T_2 - t_3\}$. We want to choose a threshold as small as possible to minimize matches that do not actually correspond to automated posts, but while still minimizing the missed automated posts. First, we discard all timestamp differences larger than 30 s, as a rough estimate of an upper bound for the maximal delay between automated posts. Within the remaining timestamp differences, we found that 85% are lower than 5 s. We conclude that a threshold of 5 s ensures to miss at most 15% of the automated posts (we suggest that the actual percentage of missed automated posts will actually be much lower since some fast users can write non-automated posts within 5-30 s). Moreover, we observed that, out of all the GT users that have at least one post at an interval less than 30 s (which is 13 users), all of them have also one post at an interval less than 5 s.

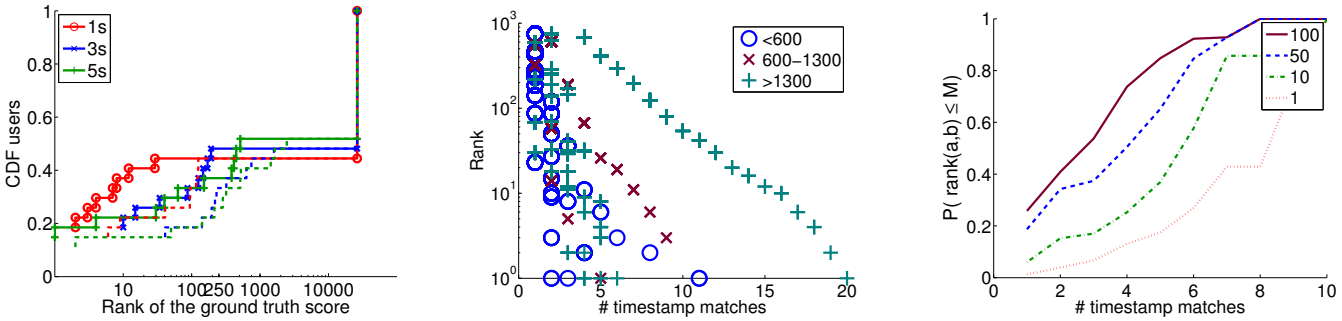
We simply define here the similarity metric between two accounts as the number of timestamp matches (within a chosen threshold). We also performed the analysis with a similarity metric defined as the number of timestamp matches divided by the cardinality of the set of timestamp differences $td(a,b)$, but the results were not as good so we will not present them here.

Figure 3a shows the CDF of the rank of the GT scores for different thresholds. We can see that for half of the users the rank of the score is low (i.e., good). We even have perfect matchings (i.e., the similarity score for the GT pair is the highest) for a quarter of them. For the second half of users the ranks are very high because they do not have any timestamp match. We think that these users are not using automated posting applications.Comparing the CDFs for the different thresholds, we observe that a threshold of one second gives the best performance (i.e., the lowest rankings) for users with at least one timestamp match (within 1 s). This is because it gives the lowest probability to have posts that match "by chance" in the dataset. However, compared to a threshold of 3 s, it misses some users for which the smallest timestamp difference of a matching post is 1-3 s. Those users appear to have no match and then get the highest ranking. This illustrates for the entire dataset the trade-off for GT users: a low threshold minimizes erroneous timestamp matches but increases the number of missed matching posts.



(a) $\#z_a \leq 10$ zip codes (b) $10 < \#z_a \leq 100$ zip codes (c) $\#z_a > 100$ zip codes

Figure 2: Probability of the $rank(a, b)$ of a pair of accounts to be smaller than 50, 250, 500 and 1000 as a function of the number of common zip codes between two accounts.



(a) Rank of the GT pairs.

(b) Pair rank as a function of the number of timestamp matches for all pairs for different number of Flickr photos (< 600, $600 - 1,300$ and $> 1,300$).

(c) Probability that the $rank(a, b)$ of a pair of accounts is smaller than 1, 10, 50 and 100 as a function of the number of timestamp matches.

Figure 3: Results for matching based on timestamps.

Implications: We conclude that if one uses applications that trigger automated postings, we can link their accounts with high confidence, we have a perfect matching for 20% of our $GT_{BayArea}$ users for a threshold of one second. As for the location profiles, we investigate the correlation between simple properties of two accounts such as the number of timestamp matches with the ranking of the score, Figure 3b. The different symbols correspond to pairs that have less thank 600, between 600 and 1300 and more than 1300 Flickr photos. We see: (i) a good correlation between the number of timestamp matches and the rank; and (ii) for the same number of timestamp matches, a Flickr account with more timestamps will have a worse rank than an account with fewer timestamps. Figure 3c shows the probability to narrow down to a set of 1, 10, 50 and 100 users as a function of the number of timestamp matches. We see that, for instance, even with only 4 timestamp matches, the set of possible matching accounts can be narrowed down to 50 accounts with probability 50%. This implies that even if people are not frequent users of such applications (e.g, they might have just tested one briefly) they still match well. In other words, automated postings are generally a clear giveaway as the probability that timestamps match closely just by chance is low.

5. LANGUAGE PROFILES

Textual metadata is the final type of feature that we consider for correlating accounts. This approach builds on existing work demonstrating that free-form text can exhibit characteristics sufficiently unique to identify an author. To explore this potential, we examined correlating Yelp reviews with Twitter postings.[6] Reviews tend to consist of multiple paragraphs of text, and for each Twitter account we considered the joint set of all its tweets, typically containing one or two sentences each. A challenge here, however, is that the same user may adopt drastically different kinds of textual structure when writing Yelp reviews (typically complete paragraphs using words mostly found in the English lexicon) versus when tweeting (typically short sentences with fewer standard words).

We found an average of 18,307 words amongst all tweets per Twitter user, and 4,153 words amongst all reviews per Yelp user, amongst $GT_{BayArea}$ and $SN_2^{BayArea}$ datasets. Furthermore, we found a total of 13,556 distinct words amongst all Yelp reviews, and 11,025,682 among all tweets. To account for the large averages and remove common words without much discriminative power, we applied a simple filter by creating a list of words to keep, and discarding the rest of

[6]We skip Flickr for this analysis as its images come with only few textual metadata in comparison to two other networks.

the words. To create the list, we determined the top 1000 most frequent words amongst the Yelp reviews. Discarding the top 1000 words gave us a list of 12,556 words. Note that the reason that the text from Yelp reviews (as opposed to tweets) are used to create the list is that many of the words in tweets are not common, and do not occur amongst Yelp reviews. After applying the filtering, we find the Yelp and Twitter users' averages reduced to 721 and 3528 words, respectively.

We then built probabilistic language models for each Twitter user by constructing normalized word histograms per user, and computed the likelihood of words from Yelp reviews to the Twitter language models. We generally followed the approaches implemented in [9]. We chose words as the unit for our models because initial experiments showed no further improvements when broadening to higher n-grams (i.e., multi-words). This is likely because *(i)* the stop list already removes what often links words together, and *(ii)* tweets consist mostly of keywords with fewer stylistic expressions.

However, examining the correlation results, we did not find the language-based approach to be very effective on its own. We omit the corresponding plot, but we saw that only 10% of ground truth users rank within the top 1000. We obtained an EER of 29.8% and a miss rate of 94.7% (at 1% false alarms). The miss rate is higher than those obtained for the location-based approaches. While the results suggest that the language-based approach is less effective standalone compared to the other approaches, we see in the next section that it is nevertheless effective when used in combination.

6. COMBINING FEATURES

The previous sections discuss matching accounts with scores computed from *individual* features (location, timing, and language). We now turn to exploiting them all *simultaneously*. The premise here is that combining the individual metrics should *(i)* achieve stronger correlation by leveraging their respective strengths, while *(ii)* making it harder for users to defend against such attacks.

6.1 Approach

We investigate unsupervised combinations of the individual scores. Specifically, we consider two approaches. First, we *average* the scores as a simple method to join them in an unsupervised fashion. Second, we examine an unsupervised combination approach, based on a Gaussian noise model, whose parameters are estimated via maximum-likelihood (ML). This method weights the contributions of the individual scores by attempting to account for the fact that the individual scores provide different levels of influence and discriminative power. As we use a standard approach for deriving the estimator, we skip further discussion here and refer to Appendix C for the details.

6.2 Results

The similarity estimators we use for the combinations for Yelp and Twitter are are: (a) zip code (b) longitude/latitude clusters, and (c) text. For Flickr and Twitter, we use: (a) zip code (b) longitude/latitude clusters, and (c) timestamps. For the zip code estimators, we also consider the contributions of the 2 similarity weights.

The CDFs of the ground truth user's ranks are shown in Figures 4a and 4b, respectively. In the CDF plots, we use the ML approach for combination, which performs better in general than averaging. For both site combinations, the CDFs provide evidence for the power of combination: the curves for joined similarity estimators are generally higher than for baseline (individual) ones.

The EER and miss rate verification metrics are shown in Tables 3 and 4 for Yelp to Twitter and Flickr to Twitter, respectively. With either measures, lower values imply better similarities. For the miss rate, we again assume 1% false alarm rate.

Table 3: *Results for Yelp to Twitter using EER and miss rates at 1% FA rates. With either measure, lower values imply better similarities. w*1 *and w*2 *denote the similarity weights 1 and 2, respectively*

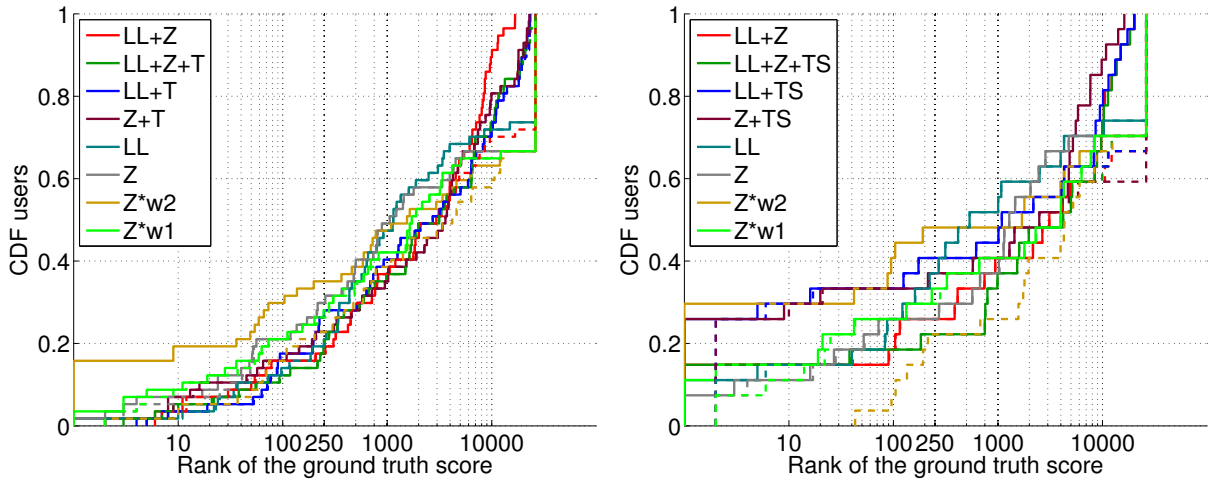
SNo	Feature/Method	EER(%)	Miss rate at 1% FA
Scores from each feature			
1	Zip code (Z)	32.3	70.2
2	Long/Lat (LL)	28.1	73.7
3	Text (T)	29.8	94.7
4	Zip code * w1	32.2	91.2
5	Zipcode * w2	32.2	85.9
Pairwise combination of features			
4	Avg (LL, Z)	24.6	71.9
5	Avg (LL, T)	28.1	73.7
6	Avg (Z, T)	31.6	70.2
7	ML (LL, Z)	29.8	70.1
8	ML (LL, T)	28.0	70.2
9	ML (Z, T)	32.6	70.1
Combination of Zip code, Long/Lat, and Text			
10	Avg	24.6	71.9
11	ML	29.8	68.4

For Yelp to Twitter, we see in Table 3 that combination also yields better miss rates than the baselines. For example, averaging yields an absolute improvement of 2% miss rate over the best estimators (zip codes and longitude/latitude), and ML yields an absolute improvement of 5% miss rate. The best result (68.4% miss rate) comes from ML combination of zip code, longitude/latitude, and text. For Flickr to Twitter correlation, Table 4 also shows that in general, combining approaches such as longitude/latitude clusters and zip code, along with timestamp, gives better results than the longitude/latitude and zip code standalone. The best result (44.4% miss rate) comes from ML combination of longitude/latitude and timestamp.

6.3 Implications

Our results show that both combination methods can yield significant improvements over standalone scores, providing evidence that attackers may leverage such techniques to correlate accounts. In the following, we explore a specific example to understand how much information it takes to successfully correlate users with the discussed methods.

Consider again the scenario of attempting to correlate Flickr and Twitter accounts from employees of a small company, but with 20 employees instead of 10, each having a Flickr and Twitter account. There are hence a total of 400 Flickr vs Twitter account pairs, among which 20 correspond to matches between the users. Hence, there are 380 pairs with mis-matched users, which means that roughly 4 of those scores fall above the 1% false alarm scoring thresh-



(a) CDFs for Yelp to Twitter.

(b) CDFs for Flickr to Twitter.

Figure 4: CDFs of the rank of the ground truth scores for similarity scores for standalone and maximum likelihood-combined approaches for Yelp to Twitter, and Flickr to Twitter. The approaches are denoted as follows: long/lat clusters (LL), zip code (Z), text (T), timestamp (TS), zip code multiplied by similarity weight 1 (Z*w1), zip code multiplied by similarity weight 2 (Z*w2).

Table 4: *Results for Flickr to Twitter using EER and miss rates at 1% FA rates. With either measure, lower values imply better similarities.*

SNo	Feature/Method	EER	Miss rate at 1% FA
Scores from each feature			
1	Zip code (Z)	29.6	85.2
2	Long/Lat (LL)	31.8	66.7
3	Timestamp (TS)	†	†
1a	Zip code * w1	38.3	85.1
1b	Zip code * w2	37.9	70.3
Pairwise combinations of features			
4	Avg (LL, Z)	29.6	77.7
5	Avg (LL, TS)	28.9	62.9
6	Avg (Z, TS)	28.5	77.7
7	ML (LL, Z)	33.3	77.7
8	ML (LL, TS)	28.9	44.4
9	ML (Z, TS)	28.5	51.8
Combination of Zip code, Long/Lat, and Timestamp			
10	Avg	29.6	70.3
11	ML	29.6	81.4

† Note that the score distributions for the timestamps did not allow for the setting of appropriate scoring thresholds for EER and miss rate computation

old for Flickr to Twitter correlation. If we consider the 44.4% miss rate at the 1% false alarm rate (our best miss rate for Flickr to Twitter correlation, obtained using longitude/latitude clusters and timestamps), we see that roughly 9 out of the 20 scores with matched users fall *below* the scoring threshold, while 11 fall above the threshold. Hence, if we only consider scores above the 1% false alarm threshold, then 11 of them are matches, while 4 are mismatches. This means that given such a 20-user scenario, with 400 Flickr vs Twitter account pairs, 75% (11 out of 15) of the account pairs with scores above the threshold are same-user pairs.

This analysis provides perspective on the types of scenarios that allow our fairly straight-forward correlation techniques to succeed. We note, however, that our main contribution concerns not the specific numbers—which we expect will only improve as correlation techniques advance, and more data becomes accessible—but the fact that a significant proportion of the users of social networks can potentially be de-anonymized with such cross-site correlation techniques.

7. RELATED WORK

A variety of efforts have examined aspects of information leakage related to our work, however none of it exploits implicit activity features attached to the content. Most closely related is a recent series of work aimed at identifying users across different social networks, similar in spirit to what we discuss yet with different approaches. Perito et al. [10] explored linking user profiles by looking at the entropy of their usernames. Irani et al. [11] studied finding further accounts of a user by applying a set of simple heuristics to its name. Balduzzi et al. [12] correlate accounts on different social networks by exploiting the friend finder mechanism with a list of 10 million email addresses. While a straightforward way to correlate accounts, most social networks have since limited the number of e-mail addresses that one can query, thus rendering this attack no longer usable on a large scale. Iofciu et al. [13] used tags to identify users across social tagging systems such as Delicious, StumbleUpon and Flickr. The authors of [14] show that group memberships present on many social networks can uniquely identify users; they leverage this to identify users visiting malicious web sites by matching their browser history against groups on social sites.

In another line of work, researchers used publicly available information from a social network site to infer specifics

about its users, without however correlating it with further accounts elsewhere. Hecht et al. [15] derived user locations from tweets using basic machine learning techniques that associated tweets with geotagged articles on Wikipedia. Similarly, Kinsella et al. [6] leveraged tweets with geotags to build language models for specific places; they found that their model can predict country, state and city with similar performance as IP geolocation, and zip code with much higher accuracy. Crandall et al. [5] located Flickr photos by identifying landmarks via visual, temporal and textual features. Chaabane et al. [16] leverage interests and likes on Facebook to infer otherwise hidden information about users, such as gender, relationship status, and age. Further similar work includes [17, 18].

Language models have been used for data de-anonymization. For example, Nanavati et al.[7] used language distribution at the n-gram level to de-anonymize reviews in an anonymous review process. A recent study showed how text posted on blogs can be de-anonymized [19]. More generally, a number of de-anonymization efforts demonstrated the power of correlation. Sweeney [20] de-anonymized medical records with the help of external auxiliary information. Likewise, Narayanan et al. de-anonymized Netflix movie ratings. In [21], a similar approach attacks a social network graph by correlating it with known identities on another. On a more fundamental level, Bishop et al. [22] discuss the need to consider external knowledge when when sanitizing a data set.

8. CONCLUSION

In this work we present a set of experiments that use straightforward data mining techniques to correlate user accounts across social networks, based on otherwise innocuous information like time patterns or location of the posts. Our approaches work independent of standard privacy measures, such as disabling tracking cookies or using anonymizing proxies. They demonstrate that tracking users just by their posting activity is a real threat. As such, the privacy implications of our results are two-fold. First, we point out that it is the *aggregate* set of a user's complete online footprint that needs protection, not just content on individual sites. Second, we find that it is hard to defend against such attacks as it is the very activity one *wants* to publish that enables correlation to succeed.

While our work remains a case study for a particular setting, it demonstrates the potential of cross-site correlation. Our approaches are conceptually simple, yet we predict that we will soon see more sophisticated—and further automated—variants in the wild that will exploit the increasing volume of user information that social networks now offer via convenient APIs. In particular, we expect that automated content analysis technology—such as face recognizers and natural language processing—will enable correlations much beyond what we demonstrate in this work.

From a research perspective, we encourage our community to devise novel privacy protections that take such threats into account and, where hard to prevent, at least support users in understanding their vulnerability. For example, we are investigating building tools that help users to identify a subset of their published content that contributes most to cross-site attacks, allowing them to modify or even delete those parts as necessary.

9. REFERENCES

[1] Social Intelligence Corp., http://www.socialintel.com/.

[2] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, ser. SP '08, 2008, pp. 111–125.

[3] "Yahoo! placemaker," http://developer.yahoo.com/geo/placemaker/.

[4] "geonames.org," http://geonames.org.

[5] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, "Mapping the world's photos," in *Proceedings of the 18th international conference on World Wide Web*, ser. WWW '09, 2009, pp. 761–770.

[6] S. Kinsella, V. Murdock, and N. O'Hare, ""i'm eating a sandwich in glasgow": modeling locations with tweets," in *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, ser. SMUC '11, 2011, pp. 61–68.

[7] M. Nanavati, N. Taylor, W. Aiello, and A. Warfield, "Herbert west: deanonymizer," in *Proceedings of the 6th USENIX conference on Hot topics in security*, ser. HotSec '11, 2011.

[8] S.-h. Cha, "Comprehensive survey on distance / similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[9] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Proceedings International Conference on Spoken Language Processing*, November 2002, pp. 257–286.

[10] D. Perito, C. Castelluccia, M. Ali Kâafar, and P. Manils, "How unique and traceable are usernames?" in *Proceedings of the 11th Privacy Enhancing Technologies Symposium*, ser. PETS'11, 2011, pp. 1–17.

[11] D. Irani, S. Webb, K. Li, and C. Pu, "Large online social footprints–an emerging threat," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, ser. CSE '09, 2009, pp. 271–276.

[12] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel, "Abusing social networks for automated user profiling," in *Proceedings of 13th International Symposium on Recent Advances in Intrusion Detection*, ser. RAID'10, 2010, pp. 422–441.

[13] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems," in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, ser. ICWSM '11, 2011, pp. 522–525.

[14] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Proceedings of the 31st IEEE Symposium on Security and Privacy*, ser. SP '10, 2010, pp. 223–238.

[15] B. Hecht, L. Hong, B. Suh, and E. H. Chi, "Tweets from justin bieber's heart: the dynamics of the location field in user profiles," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11, 2011, pp. 237–246.

[16] A. Chaabane, G. Acs, and M. A. Kaafar, "You are what you like! information leakage through users'

interests," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, ser. NDSS '12, 2012.

[17] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09, 2009, pp. 531–540.

[18] D. Gayo Avello, "All liaisons are dangerous when all your friends are known to us," in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, ser. HT '11, 2011, pp. 171–180.

[19] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *Proceedings of the 33st IEEE Symposium on Security and Privacy*, ser. SP '12, 2012, to appear.

[20] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *Journal of Law, Medicine, and Ethics*, vol. 25, no. 2–3, pp. 98–110, 1997.

[21] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, ser. SP '09, 2009, pp. 173–187.

[22] M. Bishop, J. Cummins, S. Peisert, A. Singh, B. Bhumiratana, D. Agarwal, D. Frincke, and M. Hogarth, "Relationships and data sanitization: A study in scarlet," in *Proceedings of the 2010 Workshop on New Security Paradigms*, ser. NSPW '10, 2010, pp. 151–164.

APPENDIX

A. SIMILARITY METRICS

We note, S_{IP} as the inner product metric, S_{Cos} as the cosine distance, S_{Jac} as the Jaccard distance, S_H as the Hellinger distance, d_{Euc} as the Euclidian distance, d_M as the Manhattan distance, and d_{KL} as the Kullback-Leibler divergence. Amongst these metrics, S_{IP}, S_{Cos}, S_{Jac} represent similarity metrics, where higher values denote greater similarity between distributions P and Q. The other metrics are distance metrics, where higher values denote greater dis-similarity between the distributions. P_i and Q_i represent the set of probabilities corresponding to discrete probability distributions P and Q.

$$S_{IP} = \sum_{i=1}^{d} P_i Q_i \qquad d_{Euc} = \sqrt{\sum_{i=1}^{d} |P_i - Q_i|^2}$$

$$S_{Cos} = \frac{\sum_{i=1}^{d} P_i Q_i}{\sqrt{\sum_{i=1}^{d} P_i^2}\sqrt{\sum_{i=1}^{d} Q_i^2}} \qquad d_M = \sum_{i=1}^{d} |P_i - Q_i|$$

$$S_{Jac} = \frac{\sum_{i=1}^{d} P_i Q_i}{\sum_{i=1}^{d} P_i^2 + \sum_{i=1}^{d} Q_i^2 - \sum_{i=1}^{d} P_i Q_i} \qquad d_{KL} = \sum_{i=1}^{d} P_i ln \frac{P_i}{Q_i}$$

$$d_H = \sqrt{2 \sum_{i=1}^{d} (\sqrt{P_i} - \sqrt{Q_i})^2}$$

B. SIMILARITY WEIGHTS

In §3.2, we discuss the need for adding weights to the similarity functions. The first weight is the probability of the set of common locations of two accounts (a, b) to appear between the other pairs (a, b_i), $b_i \in \widetilde{SN}_2$. For each pair (a, b) the set of common locations between their location profiles is $CL_k = LP(a) \cap LP(b)$. The unique sets of common locations from the list $(CL_k)_{k \leq N}$ are denoted by $cl_j, j = 1, \ldots, M$. We denote the frequency of a unique set of common locations by

$$f(cl_j) = \frac{\#\{k : CL_k = cl_j\}}{N}, \quad j = 1, \ldots, M,$$

and we define for each pair (a, b) the first weight

$$weight_1(a, b) = f(cl_j)^{-1}, \text{ for } j \text{ s.t. } CL_k = cl_j. \qquad (3)$$

The second weight is the product of the probability of each location in the set of common locations of two accounts (a, b) to appear in the list of common locations between all (a, b_i) pairs. All the locations that appear in the list $(CL_k)_{k \leq N}$ are denoted by $a_i, i = 1, \ldots, P$. We denote the frequency of a location by

$$f(l_i) = \frac{\#\{k : l_i \in CL_k\}}{\sum_k \#CL_k}, \quad i = 1, \ldots, P,$$

and we define for each pair (a, b) the second weight

$$weight_2(a, b) = \left(\prod_{l_i \in CL_k} f(l_i) \right)^{-1}. \qquad (4)$$

C. UNSUPERVISED COMBINATION

We propose a technique based on the assumption that the output of the *similarity estimators* can be viewed as noisy versions of an underlying ground truth similarity score; furthermore, we assume that this "noise" is Gaussian. We shall discuss this method in detail.

Let s_i^j be random variables denoting the output of jth similarity score estimator for a pair of accounts (i). We shall assume a simplistic model where we consider that these similarity scores are perturbed around a mean score s_i, with a variance σ_j^2. Furthermore, we assume the errors committed by each estimator around the true score (s_i) are Gaussian distributed with zero mean and variance (σ_j^2). More formally, $\epsilon_j \sim \mathcal{N}(0, \sigma_j^2)$, $s_i^j = s_i + \epsilon_j$,

$$p(s_i^j | s_i, \sigma_j) \quad \sim \quad \mathcal{N}(s_i^j | s_i, \sigma_j^2) \qquad (5)$$

For estimating the model parameters, we use the maximum likelihood (ML) formulation.

Let $\mathcal{D}$ denote the set of scores, and let Θ denote the set of parameters s_i, σ_j, where $i \in 1 \cdots N$, and $j \in 1 \cdots M$. Then the likelihood function can be written as,

$$\mathcal{L}(\Theta) = p(\mathcal{D}|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} p(s_i^j | s_i, \sigma_j) = \prod_{i=1}^{N} \prod_{j=1}^{M} \mathcal{N}(s_i^j, \sigma_j^2)$$

The ML estimate of the parameters can be formulated as,

$$\begin{aligned}\widehat{\Theta}_{ML} &= \arg\max_{\Theta}\big(\ln p(\mathcal{D}|\Theta)\big)\\ &= \arg\max_{\{s_i,\sigma_j\}}\sum_{i=1}^{N}\sum_{j=1}^{M}\ln\Big(\frac{1}{\sqrt{2\pi}\sigma_j}\,e^{-\frac{(s_i^j-s_i)^2}{2\sigma_j^2}}\Big)\\ &= \arg\max_{\{s_i,\sigma_j\}}-\sum_{i=1}^{N}\sum_{j=1}^{M}\Big(\frac{1}{2}\ln 2\pi\sigma_j^2+\frac{(s_i^j-s_i)^2}{2\sigma_j^2}\Big) \quad (6)\end{aligned}$$

Computing the derivatives of $\mathcal{L}$ with respect to the parameter set $\{s_i, \sigma_j\}$ and solving, we get

$$\widehat{\sigma_j^2}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(s_i^j - s_i)^2 \quad (7)$$

$$\widehat{s_i}_{ML} = \frac{\sum_{j=1}^{M}\frac{s_i^j}{\sigma_j^2}}{\sum_{j=1}^{M}\frac{1}{\sigma_j^2}} \quad (8)$$

As the parameters $\widehat{\sigma_j}$ and $\widehat{s_i}$ are coupled, we iterate the two steps (4,5) till convergence. The result is intuitive as it estimates the ground truth as a weighted combination of the output of the estimators. The weights themselves are estimated as the inverse of the variance of the estimators. We note here, that regularization can be done to prevent overfitting of the model parameters. However, for simplicity, we avoid this and use early stopping.

We found that the distributions of the similarity scores are peaky, and the second mode of these bimodal distributions are barely perceptible. To reduce the dynamic range of the similarity scores we use the transformation,

$$f(s_i^j) = \log\big(\frac{s_i^j}{1-s_i^j}\big). \quad (9)$$