Large-scale Correlation of Accounts Across Social Networks

Oana Goga[§], Daniele Perito[*], Howard Lei[ǂ], Renata Teixeira[§], and Robin Sommer[ǂɣ]

TR-13-002

April 2013

Abstract

Organizations are increasingly mining the personal data users generate as they carry out much of their day-to-day activities online. A range of new business models specifically exploit what users publish on their social network profiles, including services performing background checks and analytics providers who, e.g., associate demographics with consumer behavior. In this work we set out to understand the capabilities of machine learning techniques for linking independent accounts that users maintain on different social networks, based solely on the information people explicitly and publicly provide in their profiles. We perform a large scale study that assesses a range of correlation approaches for matching accounts between five popular social networks: Twitter, Facebook, Google+, Myspace, and Flickr. Our results show for instance that by exploiting usernames, real names, locations, and photos, we can robustly identify about 80% of the matching pairs of user accounts between any combination of two social networks among Twitter, Facebook and Google+. Our work is the first to demonstrate the feasibility of such conceptually simple privacy attacks at large scale, across several major networks, and with such efficiency.

§ Université Pierre et Marie Curie, 4 Place Jussieu, 75005 Paris, France
* University of California at Berkeley, 101 Sproul Hall, Berkeley, California 94704
ǂ International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, California, 94704
ɣ Lawrence Berkeley National Lab, 1 Cyclotron Road, Berkeley, CA 94720

This work was partially supported by funding provided to ICSI through National Science Foundation grant CNS-1065240. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

With users now performing much of their day-to-day activities online, an increasing number of companies (and governments) are already mining the data they make available online. For example, companies such as `reputation.com` mine publicly available users personal data to help them control their online reputation. On a more ethically involved note, the security contractor Raytheon just developed a software that can mine data from social networks to obtain an entire snapshot of a user's life and display it easily [2]. Such services exploit a variety of rich sources where users provide their personal information. For instance, LinkedIn reveals professional profiles, Facebook offers a view into private lives, Twitter broadcasts interests, and MySpace still remembers the past. While users often consider each of these networks a separate realm, organizations are beginning to *correlate* personas across sites to assemble a more comprehensive picture of an individual than any source alone would provide. For example, background checks for job applicants now routinely search for the accounts that a potential hire maintains [22]. Companies like PeekYou and Spokeo[1] advertise themselves as *"people search engines."* What these services provide is basically a way to list potential matching accounts starting from information like usernames and real names. However, such services operate behind closed doors. As results about correlation performance remain unknown, it proves difficult for the public to assess this emerging privacy threat.

In this work, we set out to provide answers by evaluating the real potential of this privacy threat. The main contribution of our effort is the analysis of account correlation *at scale*, using a large, real-world dataset of user accounts to derive representative results from major social networks: Facebook, Twitter, Google+, Flickr, and Myspace. The first three are among the most popular today, Flickr provides insight into a different community, and Myspace allows us to understand if one can link current profiles to an individual's past—"the Internet never forgets".

Our study builds on earlier work by Perito et al. [21] that examines the utility of different similarity metrics for matching accounts (i.e., finding pairs of accounts belonging to the same user on two social networks) by their usernames alone. Here, we extend this approach to several features, to more social networks, and to more users. For our study, we consider a set of readily available attributes that users commonly provide openly (username, real name, profile photo, and location), from which we derive classification features used to match accounts. We examine accounts of more than 200,000 users common between different social networks among the five mentioned above, obtained from crawling about 3 millions Google+ accounts. Compared to Perito's work, we find that combining features significantly improves performance over usernames alone (up to 100% improvement). Our results show for instance that when combining all features, for a false positive rate of 10^{-3}, we can expect to find about 90% of the matching pairs of accounts between any two social networks among Twitter, Facebook, Flickr and Google+, and 60% between Myspace and Twitter, Facebook or Google+. Our estimations also show that the percentage of accounts that we can match without making *any* mistakes is only about 10% lower than when allowing for the 10^{-3} false positive rate. Furthermore, for accounts that we cannot match directly between a pair of

[1] `http://www.peekyou.com/`, `http://www.spokeo.com/`

networks, we demonstrate the potential of building *correlation chains* that link matching accounts on two social networks with the help of a third. Doing so allows us to match 5% to 23% of the accounts that do not correlate directly. Finally, with our large-scale dataset we are able to show that this new multi-feature correlation indeed remains efficient at the scale of an entire social network, thanks to good performance even at extremely low false positive rates. To generalize this result, we also extend our study to a different set of users derived from an extensive list of email addresses. Compared to Google+ users, we find that the performance decreases but remains high at 60% for the Twitter/Flickr combination at 10^{-3} false positive rate (vs 85% for Google+ users). For interpretation, a 60% true positive rate means that we could successfully correlate hundreds of millions of accounts when considering entire social networks. Moreover, using this new set of users, we show that the matching classifiers can be reliably trained on the Google+ dataset with no bias.

We structure the remainder of this paper as follows. We start by describing our data in §2. We then present the features we use as well as the metrics to measure the similarity between them in §3. Our matching methodology is presented in §4 and the results for matching different social networks are presented in §5. In §6, we extend the techniques to match social networks by building correlation chains. We finish with the related works §7 , some discussions §8 and the conclusions of our results §9.

2 Data Sets

For our study we examine five major social networks: Twitter, Facebook, Google+, Flickr and Myspace. Extracting features from public profiles is straightforward. The main challenge is to obtain a ground truth, i.e a set of accounts that we *know* belong to the same user. Here we exploit the fact that Google+ allows users to explicitly list further accounts they have on other networks on their profile pages. We randomly crawled about 3 million Google+ profiles and arrived at the ground truth set summarized in Table 1, which reports the number of matching accounts for different combinations of social networks[2] [3].

To complement the Google+ dataset, we obtained a second set of ground truth users that contains 19,000 matching pairs of accounts between Twitter and Flickr (this set is disjoint from the previous one). This data was obtained exploiting the "Friend Finder" mechanism with a list of emails obtained from a spam project. The Friend Finder take as input the list of emails and outputs if the emails correspond to accounts on Twitter and Flickr. Because everybody receives spam, we consider this dataset to be representative of generic users. We call this data the email ground truth dataset and we use it in §5.4 to evaluate the representativeness of our results from the Google+ dataset. The value of the email ground truth dataset comes from the fact that it contains a representative sample of users in general, however it is much harder, and even impossible for some social networks, to obtain it the large quantities suitable for our study.

[2] The Google+ social network proves easy to crawl as it provides a comprehensive directory of all accounts, and does not block crawlers.

[3] Although our dataset allows to study the combination of Flickr with all other networks, we limit ourselves to the pair Twitter-Flickr with serve for the comparison with the email dataset.

Table 1: Number of users in the Google+ ground-truth dataset for different combinations of social networks.

Twitter - Facebook	76,332
Twitter - Google+	205,709
Twitter - Myspace	9,015
Facebook - Google+	164,333
Facebook - Myspace	9,610
Myspace - Google+	36,440
Twitter - Flickr	35,208
Twitter - Facebook - Google+	76,600
Twitter - Facebook - Myspace	4,207
Twitter - Google+ - Myspace	9,015
Facebook - Google+ - Myspace	9,610

For the accounts in the ground truth dataset we collected the public information present in the corresponding profiles on the four social networks. For Google+ and Myspace we downloaded the profile pages and extracted usernames, real names, locations, and profile photos from the HTML code. We use these four attributes because they are generally available on most social networks. For Twitter and Facebook we used their APIs to obtain the same features, except for Facebook where the API does not provide location information. We summarize the availability of features per profile in the next section. We note that all the data we use in our study is publicly available and as such generally approved by our local IRB for research usage.

3 Features and Similarity Metrics

For each account we have four attributes extracted from the users public profiles. From these attributes we derive six features that we will use to correlate accounts: username, real name, cross name, location, photo and face. To compare the similarity of these features we borrow a set of established metrics from past work in the security and multimedia community, which we describe in the following section. Our goal is to apply current state-of-the metrics to our setting.

3.1 Similarity Metrics

Name similarity. Previous work in the record linkage community has demonstrated that the *Jaro string distance* is the most suitable metric to compare similarity between real names [9]. In a more recent study Perito et al. [21] showed that the Jaro distance also works well for comparing usernames, and performs only slightly worse than more complex methods also taking name popularity into account. Note that even if the Jaro distance does not account for the popularity of the names, we can still reach very high accuracy in linking by using additional information such as photos and location. Thus, for our study we use the Jaro distance to measure the similarity between both real names and usernames. In addition we also measure the similarity using the Jaro distance between the username on one social network and the real name on the other social network and vice versa. We consider the maximum of this two distance as a new feature which we call *cross name*. This new feature catches several matching accounts omitted by both username and real name.

Photo similarity. This metric tries to identify users that are using the same photo on multiple account. However, due to image transformations, the same profile photo can come in different representations on multiple social networks. It may for example appear in different sizes and resolutions, be cropped, or have different filters applied to it. To measure the similarity of two photos while taking such modifications into account, we use *perceptual hashing*, a technique originally invented for identifying illegal copies of copyrighted content [4]. While the description of perceptual hashing techniques is outside the scope of this paper, the basics are relatively straight-forward: an image is first reduced to a "fingerprint" by extracting its salient characteristics. This fingerprints has to remain almost unaltered even if changes in the original image are introduced, such as scaling, cropping, compression or rotation. The fingerprints of different images can be compared against each other using simple string similarity metrics, like the Hamming distance. Images with very high similarity will likely be the same image.

Face similarity. While our photo similarity metric aims to find instances of the *same* photo, we also deploy a face recognizer to measure the similarity between faces to identify different photos showing the same person. To this end, we used the freely available OpenCV package [3] along with IDIAP's Torchvision package [14]. The algorithm used involves first performing facial feature extraction for all images, followed by training of a Gaussian Mixture Model (GMM) used to obtain a similarity score. A more detailed list of steps can be found in Appendix A.

Location similarity. For all accounts (except Facebook) we also have textual representations of the users' location, like the name of a city. However, as social networks use different formats for this information, we cannot just perform a textual comparison. Instead, we convert the location to latitude/longitude coordinates by submitting them to the Bing API [1]. We then compute the distance between two locations as the actual geodesic distance between the corresponding coordinates, normalizing the value into the range between zero and one.

3.2 Availability

Users do not necessarily provide all our attributes on their profile pages. Table 2a shows a general breakdown of attribute availability per network. Furthermore, Table 2b shows availability for pairs of networks, i.e., how often each attribute appears in both of a user's accounts in our ground truth dataset (recall that on Facebook we do not have location information available). Real names appear in more than 90% of Twitter, Facebook and Google+ accounts but they only appear in 19% of Myspace accounts. The availability of location depends on the social network and it varies from 26% for Myspace to 80% for Twitter. The photo availability represents the number of accounts that do not have the default avatar and varies slightly around the 90% range.

To confirm that generally our metrics can indeed separate accounts belonging to the same user from unrelated ones, we compute pair-wise similarity scores for all Twitter and Google+ profiles. Figure 1 shows the corresponding histograms per feature, separately each time for matching and non-matching pairs of accounts (zero means no similarity and one means perfect similarity). For username, real name and the cross name features, we see a clear distinction between the two distributions, suggesting that they indeed all can contribute to identifying related accounts, however the distinction is

Table 2: Legend: T = Twitter, F = Facebook, G = Google+, M = Myspace, Fl = Flickr; † for accounts in the email ground truth dataset.

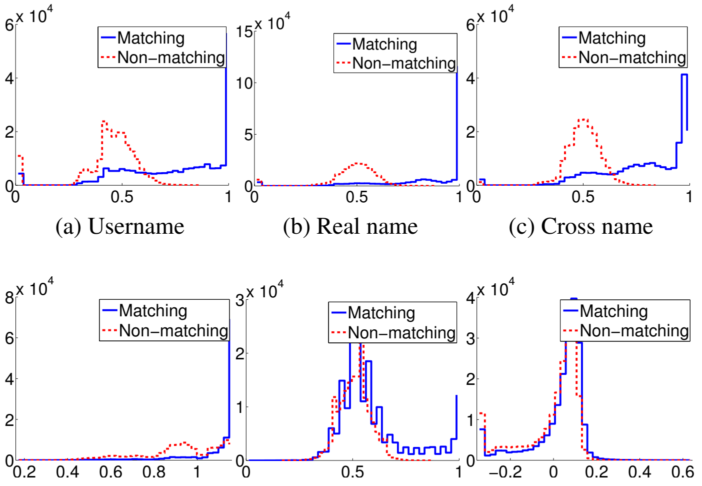
(a) Attribute availability per network.

	Twitter	Facebook	Google+	Myspace	Flickr	Twitter†	Flickr†
Username	100%	100%	100%	100%	100%	100%	100%
Real name	91%	97%	93%	19%	75%	95%	30%
Location	80%	0%	69%	26%	53%	58%	11%
Photo	95%	97%	89%	91%	94%	99%	29%

(b) Attribute availability for pairs of networks.

	T - F	T - G	F - G	M - T	M - F	M - G	T - Fl	T-Fl†
Username	100%	100%	100%	100%	100%	100%	100%	100%
Real name	98%	98%	99%	15%	17%	20%	75%	30%
Location	0%	55%	0%	23%	0%	22%	47%	8%
Photo	94%	90%	89%	85%	89%	75%	92%	28%

less clear for location, photo and face. We will explain in the next sections the reason why these latter attributes are still important for account correlation.



(a) Username (b) Real name (c) Cross name

(d) Location (e) Photo (f) Face

Fig. 1: Histograms of similarity scores for matching and non-matching pairs.

4 Matching Methodology

Given two sets of accounts from different social networks, our goal is to find pairs that belong to the same individual. We approach this task as a classification problem: we train a binary classifier with similarity scores for matching and non-matching account pairs from our ground truth set, and then use the resulting models to predict matches on new data. We detail our methodology further below, including a discussion of classifiers we use and training challenges that our setting imposes.

4.1 Classification Strategy

For each pair of accounts the classifier takes as input the similarity scores for each feature, and outputs probabilities that accounts match. We interpret the probabilities as an overall score of account similarity, and then select a cut-off threshold to separate matching from non-matching pairs. Choosing a specific threshold constitutes the standard trade-off between the proportion of pairs that we classify correctly (the *true-positive rate*) vs. incorrectly (the *false-positive rate*). To evaluate the classifier's performance, we split our ground truth data into training and testing sets using 10-fold cross validation.

Since we focus on matching accounts at large scale it is imperative to tune for small false positive rates. For illustration, assume a classifier operating at 50% true positive rate for 10^{-4} false positives. If we have only 1000 accounts on one social network that we want to match with 1000 accounts on another, the matching algorithm will, on average, return a list of 500 matching accounts and 100 non-matching accounts (since there are 10^6 pairs in total); assuming each account on the first network has exactly one match on the second. In this situation, that might indeed constitute a reasonable result as the number of true positives is five times higher than the false positives. However, if instead we have 10 million accounts on each network, the matching algorithm will return 10^{10} false positives for just $5 \cdot 10^6$ correct matches, i.e., a few orders of magnitude more. To come to a useful result in this situation, we need to instead tune the false positive rate down to 10^{-7} or 10^{-8} for example.

Following standard practice, we use ROC curves in the remainder of the paper to examine this trade-off, focussing on regions with low false-positive rates suitable for our setting. A ROC curve is a simple representation of the relationship between false positives and false negatives for different thresholds in a classifier. When using ROC curves, it is important to take confidence intervals into accounts for all observations: for two curves to differ significantly, these intervals must not overlap [13]. To compute 95% confidence intervals of each point in the ROC curve we use the threshold averaging method [10], which we choose for its ability to report confidence for both true and false positive rates. The method works as follows: for each classifier, the 10-fold cross validation gives posterior probabilities that we use to generate 10 separate ROC curves. The algorithm first randomly selects a subset of all thresholds used to generate the 10 ROC curves. Then, for each of these thresholds, it estimates the corresponding points on the 10 ROC curves. Finally, from these points it determines the median and standard deviation for true/false positive rates and uses them to derive the confidence intervals.

4.2 Training Challenges

Our setting poses two challenges for applying standard classifiers. Firstly, we face a large class imbalance as the number of possibly matching accounts is much lower than the number of non-matches. Note that this imbalance stems from the nature of the problem—not the way data is gathered—and thus carries important information about prior class probability. To handle the imbalance, we train the classifier with all pairs but assign higher weights to the matches, inversely proportional to their ratio in the training set. Doing so prevents the classifier from predicting all the pairs of accounts as non-matches. The second problem concerns features missing in some user profiles. As discussed above, users may choose not to publish their location or forget to upload a profile photo. We assume they are missing at random, and hence do not present an opportunity for account correlation themselves. However, we must ensure that our classifiers treat such missing features robustly. In the following subsection, we discuss the types of classifiers we examine and how they handle such missing data.

4.3 Classifiers

For our study, we examine the following types of classifiers.

Naive Bayes decides if two accounts match based on the probability that each feature's similarity score belongs to the matching class, assuming that the distribution of feature scores in each class is based on a kernel density estimation. The Naive Bayes classifier has a natural way of handling missing values of a feature: during training, feature instances with missing values will not be included in the feature-value-class probability computation. During testing, if a particular user has a missing feature value, then that feature will be omitted from the prediction calculation. Hence, the computation of the predicted-class will not always use the same set of features for all pairs of accounts, but this does not bias the computation in any way.

Decision Trees decide if two accounts match by traversing a tree of questions until they reach a leaf node; the leaf node then specifies the result. In our setting each node represents a threshold for a given feature; the classifier tests the input account against that value and takes the appropriate branch. The most popular way to handle missing features is at training time to only create branches on present values, and at testing to take all the branches of the node representing the feature whose value is missing and then select the class with the highest frequency among the leafs. Decision Trees prove useful for eliminating redundant features, and they allow to directly interpret results by following the decision process. The drawback is that the decision boundaries are rough because Decision Trees can only make horizontal and vertical splits.

Logistic Regression is a linear classifier that bases its decisions on a linear combination of all the similarity scores of each feature. Logistic Regression does not have a native way to handle missing values, so they must be imputed. The most common way is to replace them with the median or mean of all existing feature values. We tested both methods and the imputation with the median value gives better results.

SVM is a large margin classifier that obtains the decision boundary with the largest distance between matching and non-matching observations. Boundaries can either be linear or not. Missing values are imputed in the same way as for Logistic Regression.

Figure 2 compares the performance of different classifiers in terms of false vs. true positive rates, trained on all features. The Naive Bayes and SVM classifiers (both linear and kernel) perform the best, Logistic Regression is close to the first two while Decision Trees exhibit the lowest performance for small false positive rates because of its rough decision boundaries. Since Naive Bayes takes much less time to compute than SVM on our large data set, we use it for computing all results for the remainder of this paper.

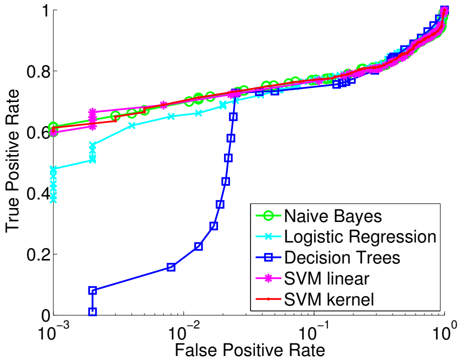
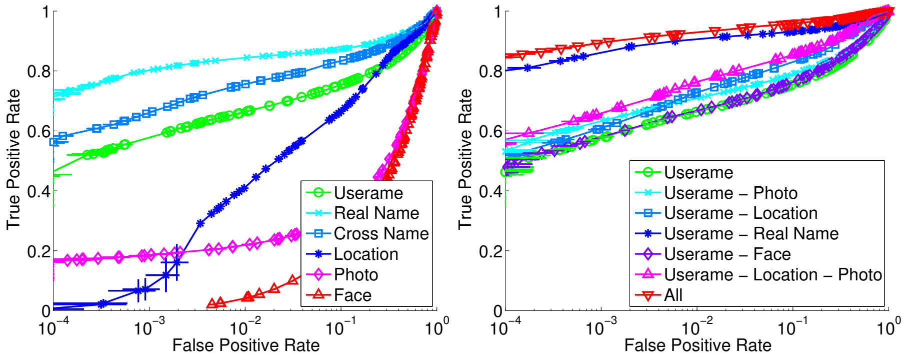


Fig. 2: Comparison of the performance of different classifiers.

5 Matching Results

This section presents our matching results. We start by investigating each feature's matching performance individually in §5.1, and then study different combinations in §5.2. Next, we change perspective to understand the number of accounts these results allow us to match successfully between different social networks (§5.3). We discuss how the results from our Google+ dataset are representative of the whole population in §5.4 and we finish the section by investigating how our results generalize to very small false positive rates suitable to match entire social networks (§5.5).

5.1 Examining Features Individually



(a) Matching performance based on individual features.

(b) Improvement over username when adding photo, location, face, and real name information.

Fig. 3: Performance of matching Twitter to Google+.

Figure 3a shows the ability of individual features to predict matching account pairs, obtained by building a separate classifier for each feature alone. For testing we use all the matching pairs we have in the ground truth set (e.g 76,332 for Twitter to Facebook correlations) and an equal part of non-matching pairs that we randomly select from all possible non-matching pairs (§5.5 will shows results when testing with all non-matching pairs). The Figures 3a and 3b show the true vs. the false positive rates for matching Twitter to Google+ accounts but the observations we derive hold on matching other social networks as well (we will take a closer look at them in §5.3). The x-axis is in log scale to concentrate on small values. The vertical and horizontal lines represent the 95% confidence intervals. Note that the vertical confidence intervals are very small and barely noticeable on the plots and the horizontal confidence interval are very large at the beginning of the ROC curve but quickly decrease afterwards. They are larger at the beginning because we do not have many observations for this ranges of false positive rates. Throughout the next four subsections whenever we give a true positive rate without further specifics, we define that as meaning 10^{-3} false positive rate. We explain how this results extend to much smaller false positive rates such as 10^{-8} in §5.5. We limit ourselves to higher false positive rates in this section because estimating false positive rates such as 10^{-8} requires a lot of resources and time and it is prohibitive to do for all the classifiers we present here.

The plot compares the performance for matching accounts using *username*, *real name*, *cross name*, *location*, *photo* and *face*. The three features involving names (*username*, *real name*, *cross name*) perform best among all. Among them, *real names* scores highest, achieving around 80% true positive rate for a 10^{-3} false positive rate. At a first look *photo* seems to be a bad predictor alone; however upon inspecting the data we found that there are just too few accounts sharing the same image in these two networks (about 20% of accounts). However, for the accounts which do, *photo* does predict matches with high accuracy. As expected, *location* is not a good predictor by itself; finding two accounts at the same place rarely means they both belong to a single individual. *Face* performs the worst probably because our face detection algorithm has a bad performance since it is only trained with one photo for each user. We believe that for the face to be a more effective feature, it must be trained with more photos.

We define the *discriminability* as the property of a feature to have a similarity threshold which can split matching and non-matching pairs of accounts with a very high accuracy. A high discriminability is the single most important property that a feature needs to be able to match alone accounts at scale on different social networks. For example, *photo* is a good predictor alone because it has a very high discriminability for high similarity scores. Out of all the similarity scores between all accounts, there are zero non-matching pairs with similarity scores higher than 0.7 while there are 18% of matching pairs with such scores. This means that for scores higher than 0.7 the *photo* can discriminate perfectly between matching and non-matching pairs. Even though we cannot match a large percentages of accounts with *photo* alone, we can however match a small percentage but with a high confidence (thus at scale). As we expect *username*, *real name* and *cross name* have a very good discriminability for high similarity scores. For example for *username*, there are zero non-matching pairs with similarity scores higher than 0.85 wile there are 42% of the matching pairs with such values. On the other hand for *location*, we still have 10% of non-matching accounts with similarity scores higher than 0.99 and even 2% with similarity scores of 1. Therefore there is no similarity sores that can perfectly split matching and non-matching accounts with *location*. *Face* is in the same category as *location*.

Features can be either *strong* (high discriminability) such as *photo* and *username* or *weak* (low discriminability) such as *location* and *face*. Other possible strong features could be face detection based on multiple photos, location pattern similarity extracted from all the posts, similarity between friends or any other feature that, when present, can make a user unique. Other weak features can be religion, gender, employer or any other descriptive attribute provided by a user in their profile that is not specific to him.

5.2 Power of Combining Features

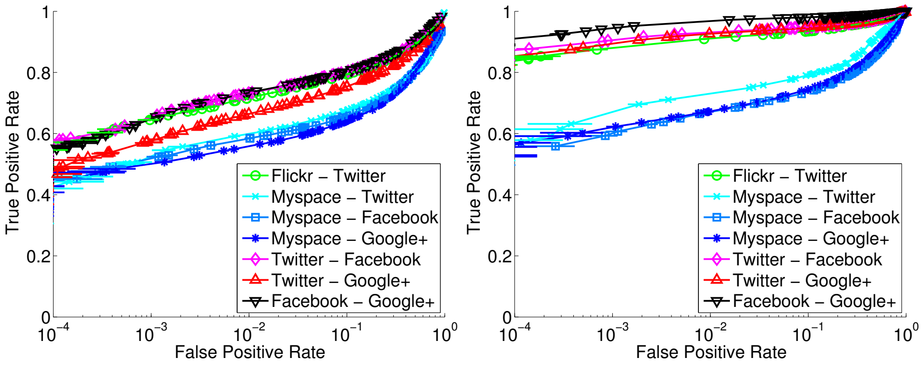
To assess the power of using features jointly, we proceed by training classifiers for each possible combination. Figure 3b presents the true vs. false positive rates for what we consider the most interesting instance: the extent to which further features can improve the performance of one of the best individual classifiers, usernames. Here we choose username as the base since of all the name-based features—which generally work best individually—it represents the one always available. The figure shows *usernames* alone; combined with *photo*, *location*, *face*, *real name*, *location/photo*; and finally all features. We see that combining *usernames* with *real names* yields to the largest individual improvement. Adding *photo* or *location* improves performance significantly as well; even more so in combination. On the other hand, using *face* does not lead to much of a contribution. Taking all features together, their combination achieves as much as 90% true positive rate at 10^{-3} false positive rate.

When combining multiple features together, an important characteristic that shows the potential of the combination is the complementarity of the features. The *complementarity* is a characteristic that says if two features detect the same or separate sets of accounts. For example, if we choose a threshold corresponding to a 10^{-3} false positive rate, 36% accounts that are detected by *photo* are not detected by *username* and 15% are not detected by *real name*, thus *photo* and *username* or *real name* are complementary features. This explains the improvement when they are combined together. Also, *username* and *real name* are complementary as 14% accounts detected by *username* are not detected by *real name* and 38% the other way around.

When we combine a weak and a strong feature, the weak feature plays the role of filter or booster for the strong feature. For *username* and *location*, if two users have similar usernames but they are in separate parts of the world we want them to be classified as non-matching accounts. Thus *location* plays the role of filter for *username*. In the case where two users have slightly different usernames but they live in the same place, we would like the classifier to boost a little their similarity score. Indeed, there are pairs in our dataset that are not matched by *username* alone because they have similarity scores slightly below the threshold, but they can be matched when adding *location*. Thus, features such as *location* when used in combination with strong features can improves the overall performance of classification but they can only slightly alter the scores obtained by the strong feature. If the username similarity is not already close to the threshold, *location* cannot do much. However, if we combine two features that are discriminative (and complementary), for example *username* and *photo*, the classifier can detect matching accounts that have very low similarity between usernames if they have high similarity between the photos. Thus, when combining two strong features we can detect accounts that would have never been detected by the other strong feature (even if this strong feature is combined with other weak features). When matching social networks it is a good idea to combine both strong and weak features as they can detect different kinds of users.

5.3 Matching Across Social Networks

In this section we compare the performance of matching accounts across Facebook, Google+, Flickr, Myspace and Twitter. The correlation performance varies between these social networks; in this section we present the differences and examine their root causes. Figure 4 presents the ROC curves for matching all pairs of networks using *username* in 4a, and all features in 4b. We present them as we think they are the most interesting because *usernames* are always present, and all features give the best performance (see previous section). However we discuss in the text other features as well.



(a) Username

(b) All features

Fig. 4: Matching performance across social networks

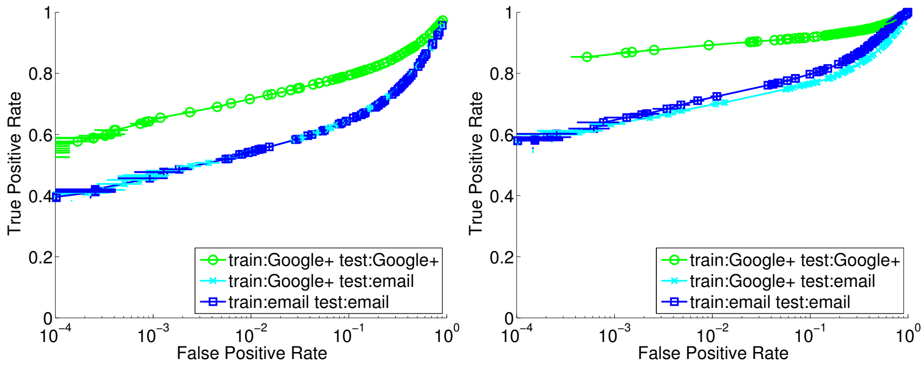
Figure 4a shows that we can match any pair using just *username* with a true positive rate around 50-60%, suggesting that the overall username similarity is quite uniform across different social networks. The true positive rate proves better for combinations that do not include Myspace because on the older networks, users tended to use aliases as their usernames. However, even for Myspace we can reliably match 50% of all accounts for any combination. When matching with *real names*, we likewise see that pairs including Myspace perform worse than others. Correlating Facebook with Google+ yields the best results, which is unsurprising as these networks expects users not to use pseudonyms. Facebook to Twitter and Twitter to Google+ performs slightly worse because Twitter accepts aliases. When matching on *photo* alone we find the best true positive rate between Twitter and Google+, followed by Myspace and Twitter, and Twitter and Facebook; while the worst is Myspace to Facebook. Photos show different matching patterns than names: users more often share the same photo between Twitter and Google+, and Myspace and Google+. For *location*, users behave more consistently between Myspace and Google+ than Myspace and Twitter, or Twitter and Google+.

When we combine all the features together, we observe that generally matching of pairs that do not include Myspace again perform much better than others, see Figure 4b.

Matching Twitter to Facebook, Facebook to Google+, and Twitter to Google+ attains true positive rates in the range of 90%; while Myspace to Twitter, Myspace to Facebook, and Myspace to Google+ reaches only 60% true positives. This gap is actually caused by the availability of real names on Myspace, as only 19% of Myspace users have specified their real name in their profiles and the *real name/username* combination contributes the most to the true positive rate when we combine all the features. However, we deem even 60% a high success rate for matching across two social networks.

5.4 Representativeness of Results

As we collected our ground truth dataset from Google+ where people voluntarily publish links to their accounts on other social networks, our results might not be representative for a more general user population. To investigate this effect, we cross-check the results with the email ground truth dataset (see §2).



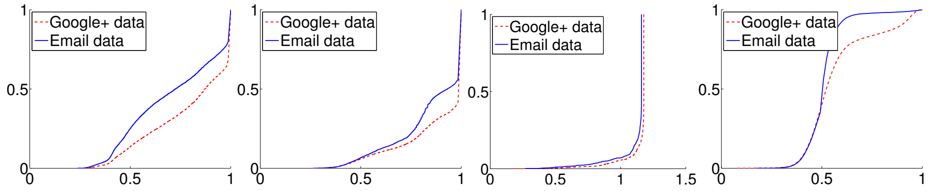
(a) Username

(b) All features

Fig. 5: Matching performance for Google+ and email ground truth users.

The question we aim to answer concerns whether the classification results (ROC curves) remain similar when we train with the email data, compared to using the Google+ data for that. Figures 5a and 5b show the ROC curves for matching Twitter to Flickr accounts for classifiers both trained and tested with Google+ data, classifiers trained with Google+ data but tested with email data, and classifiers both trained and tested with email data. As we can see in Figures 5a and 5b the ROC curves for classifiers trained with Google+ data/email data and tested on email data are the same. This suggests that the Google+ data does indeed not introduce a bias into the classification and thus we can confidently train classifiers with ground truth data from Google+ to match accounts on different social networks.

Another observation from Figures 5a and 5b is that the overall true positive rate of matching accounts from the Google+ dataset is higher than the one from the email dataset. For a false positive rate of 10^{-3} we can see a difference of 20% between Google+ and email users when matching on *username*, and 25% when matching on all features. The difference has two reasons. First, Google+ users have more similar usernames between their Twitter and Flickr accounts, and second the availability of other features is higher for Google+ than for email users; see Table 2b. Figure 6 shows the CDFs of the similarity scores between username, real name, location and photo for accounts in the Google+ and email dataset. For example, only 41% of email users have a Jaro distance between usernames higher than 0.8 while 58% of Google+ users do.



(a) Username (b) Real name (c) Location (d) Photo

Fig. 6: Comparison of CDFs of similarity scores for the Google+ and email data.

Thus, the results in Section 5.3 for matching different social networks are tilted towards a category of users with usernames that are more similar across social networks and who put more information into their profiles. We note however that many of the users difficult to match in the email ground truth dataset never posted anything on Twitter or Flickr and are therefore not the users for which finding the matching accounts would be the most valuable.

5.5 Generalization of Results

Previous sections show how well different features and their combinations can match accounts, as well as how the matching performance varies with the social networks we consider. This section investigates how our results generalize to false positive rates sufficiently small to match entire social networks.

The ROC curves in §5.1, §5.2 and §5.3 were obtained by testing the classifiers with an equal number of positive and negative pairs of accounts (equal to the number of matching pairs we have in the ground truth set) and show true positive rates for false positive rates as low as 10^{-3} or 10^{-4}. However, if we want to match entire social networks, this false positive rates are still too high. Indeed, Facebook has recently passed 1 billion users, Twitter and Google+ have more than 500 million, and Myspace more than 250 million users.[4] A fraction 10^{-4} out of all possible non-matching pairs of accounts between Google+ and Myspace would give $10^{-4} \cdot 5\ 10^{8} \cdot 2.5\ 10^{8} \approx 10^{13}$ false matching accounts for around 10^{8} matching accounts, which is useless as the false positives are a few orders of magnitude higher than the true positives. To match entire social networks, a false positive rate of 10^{-8} would be a maximum suitable rate as it gives a similar number of false positives and true positives.

To estimate the true positive rates for such small false positive rates we have to test the classifiers with more negative (non-matching) pairs of accounts. If we test the classifiers with 10,000 negative pairs, the minimum false positive rate we can observe is 10^{-4} (10^{-3} if also estimating confidence intervals). Here, to estimate smaller false

[4] http://en.wikipedia.org/wiki/List_of_virtual_communities_with_more_than_100_million_users

positive rates, we test the classifiers with 100,000,000 negative pairs. We are able to do this because we have extensive ground truth data that allows us to have enough examples of negative pairs. Note that the number of negative pairs only impacts the range of false positive rates for which we can see accurate estimates and does not modify the rest of the ROC curve. We only present results for a subset of classifiers as testing all our classifiers with such large number of pairs is prohibitive. We study two pairs of social networks: Twitter to Myspace because it is one of the pairs that has the lowest performance, and Twitter to Facebook because it is one of the pairs that has the highest performance.

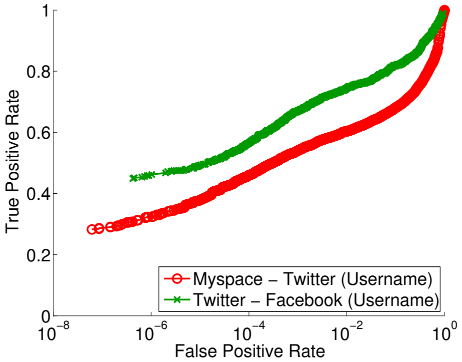


Fig. 7: ROC curves including estimates for small false positive rates.

Figure 7 shows the results obtained for Myspace–Twitter and for Twitter–Facebook correlation with a classifier using *username* alone. We observe that the ROC curve with logarithmic x-axis has an almost linear body, i.e., the true positive rate decreases linearly when the false positive rate decreases exponentially. For example, for the Myspace–Twitter matching with *usernames*, the true positive rate goes from 60% to 46% to 34% when the corresponding false positive rate goes from 10^{-2} to 10^{-4} to 10^{-6}. However, for even smaller false positive rates, the true positive rate stabilizes. At the extreme, we have a positive (and even quite high) true positive rate even for a false positive rate of zero (for instance, 27% for the Myspace–Twitter matching with *usernames*). This is due to the fact that the score distribution for non-matching pairs is exactly zero beyond some threshold smaller than one. Hence, for large enough thresholds, the classifier simply does not catch any false positive. The interesting part is that, even for thresholds so close to one, we still catch a large fraction of the true positives, especially when combining all features (see Tab. 3 below).

Looking more closely at the true positive rates for a false positive rate of zero, we observe that the estimates obtained with a large number of negative examples are actually close to the estimates obtained with a smaller number of negatives (equal to the number of positives as used in the previous sections). For Myspace to Twitter correlation, testing with 9,000 negative pairs gives a 33% true positive rate while testing with 81 million negative pairs gives a 27% true positive rate. For Twitter to Facebook correlation, testing with 76,332 negative pairs gives a 46% true positive rate while testing with 100 million negative pairs gives a 43% true positive rate. Thus, we can reliably use estimates obtained with a smaller number of negative pairs (at a reasonable computational cost) to describe the true positive rates at false positive rate of zero. Table 3 shows the results for all combinations of social networks for classifiers based on *username* alone and on all features combined. We observe that the true positive rates are very high for all pairs of social networks, especially when we combine all the features together. Matching any combination of Twitter, Facebook and Google+ stays in the 80% range while matching any combinations that include Myspace stays in the 50% range.

Table 3: True positive rates for a false positive rate of zero

Twitter	Facebook	username	46%
Twitter	Facebook	all	83%
Twitter	Google+	username	41%
Twitter	Google+	all	81%
Twitter	Myspace	username	33%
Twitter	Myspace	all	51%
Myspace	Facebook	username	39%
Myspace	Facebook	all	50%
Google+	Facebook	username	49%
Google+	Facebook	all	88%
Myspace	Google+	username	39%
Myspace	Google+	all	49%
Flickr	Twitter	username	45%
Flickr	Twitter	all	79%
Flickr	Twitter (email)	username	37%
Flickr	Twitter (email)	all	55%

We conclude with an interesting observation. As mentioned above, the similarity thresholds corresponding to very small (or zero) false positive rates are close to one. This means that only pairs of accounts that have an almost perfect matching impact the true positive rates. Consequently, even if we used similarity metrics that only measure if two usernames, real names, photos or locations are exactly the same, we would not be too far from the performance of more mismatch-tolerant techniques to measure similarity.

6 Correlation Chains

Our results so far show that we can link accounts between pairs of social networks with high confidence. For example, we can correlate Twitter with Myspace with a true positive rate of 60%, and Twitter with Facebook with a true positive rate of 90%, both for a false positive rate of 10^{-3}. We now examine if we can also match the remaining 40% and 10%, respectively, by constructing *correlation chains* that introduce a third account from a different network when we cannot match a pair directly. Figure 8 illustrates the general idea. Assume for example that the similarity between accounts 1 and 2 is lower than the threshold needed for correlating them with the classifier discussed so far, yet the similarity between accounts 1 and 3, and 3 and 2 is sufficiently high. Then we can set up a correlation chain that matches accounts 1 and 2 indirectly by going through 3.

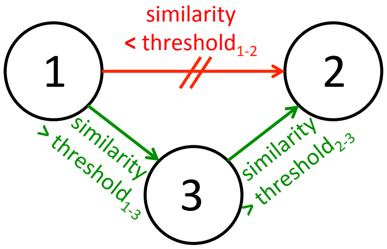


Fig. 8: Correlation chains.

If we consider the four social networks Twitter, Facebook, Myspace and Google+, we can build a total of 12 3-step correlation chains. For a correlation chain to become effective, it needs to combine at least two discriminative features. For example, if between accounts 1 and 2 the username similarity is lower than the correlation threshold, it would be almost impossible for the same feature to nevertheless work for both 1 and 3, and 3 and 2. Thus each branch of the chain $1 \rightarrow 3$ and $3 \rightarrow 2$ needs to exploit a different feature. Here, we systematically use classifiers that combine all features in order to maximize the effectiveness of correlation chains.

Table 4: Performance of correlations chains. († out of pairs that cannot be correlated directly.)

SN1	SN2	through SN3	% pairs matched using correlation chains†
Facebook	Myspace	Google	23.33%
Facebook	Myspace	Twitter	19.00%
Facebook	Twitter	Google	19.72%
Facebook	Twitter	Myspace	5.63%
Facebook	Google	Twitter	22.15%
Facebook	Google	Myspace	10.77%
Twitter	Myspace	Google	19.48%
Twitter	Myspace	Facebook	11.34%
Twitter	Google	Facebook	22.31%
Twitter	Google	Myspace	11.72%
Google	Myspace	Facebook	13.20%
Google	Myspace	Twitter	18.36%

For each pair of social networks, we choose the correlation threshold as the threshold corresponding a 10^{-3} false positive rate (this false positive rate is chosen to illustrate the correlation chains results but the method works with any false positive rate and associated threshold). For a possible chain, Table 4 shows the fraction of account pairs that are matched using the correlation chain, out of all the pairs that cannot be matched directly. For all the combinations of social networks, correlation chains can match between 6% to 23% of the total pairs that cannot be discovered directly. As a result, correlation chains increase the overall matching performance between any two social networks by a few percentage points.

7 Related Work

Our work is the first to investigate how to match accounts of a user in different social networks *at scale*. However, a number of prior studies have investigated techniques to correlate users accounts across social networks in different conditions.

Closest to our work are Perito et al.'s [21] work on exploiting the similarity between usernames to correlate accounts across social networks and Irani et al.'s [12] study of how one can find accounts of a user by applying simple modifications to her name. We extend these studies to a richer set of features and to more social networks, and we demonstrate the feasibility of such correlations at scale. A number of previous studies exploited the similarity between social network profiles to improve the ranking performance when searching for people. For example, You et al. [26] proposed to improve ranking results by linking people names on the web to their names on social networks. Motoyama et al. [16] and Bartunov et al. [7] proposed algorithms to match the contacts of a given user on different social networks. Although some of these matching approaches are similar to ours, our techniques are better suited to match entire social networks. With a different approach, Balduzzi et al. [6] correlate accounts on different social networks by exploiting the friend finder mechanism with a list of 10 million email addresses. Most sites have since limited the number of e-mail addresses that one can query making this approach unfeasible at scale. Other studies exploited different kinds of information present in users profiles to correlate their accounts. Iofciu et al. [11] used tags to identify users across social tagging systems such as Delicious, StumbleUpon and Flickr. Wondracek et al. [25] identified the users who visit malicious web sites by matching their browser history against group memberships. The technique is possible because the group membership present on many social networks can uniquely identify users. Mishari et al. showed that community reviews could be linked across different sites by exploiting the writing style of the authors [15]. Finally, Acquisti et al. demonstrated the power of face recognition algorithms by linking online and offline photos with Facebook accounts [5]. Despite the interest of these studies, the techniques proposed can hardly scale to entire social networks.

Another line of research related to our study is the work on de-anonymizing databases and graph data. Even though it is a conceptually different problem, some of the techniques are related and might be applied to match accounts on different social networks. For example, Narayanan et al. [20] proposed an algorithm that can de-anonymize the graph of a social network using the graph of another social network as auxiliary information. Two other studies showed that text posted on blogs can be de-anonymized [18], [17]. Srivatsa et al. explored how mobility traces can be de-anonymized by correlating their contact graph with the graph of a social network [23]. Sweeney [24] de-anonymized medical records with the help of external auxiliary information and Narayanan et al. de-anonymized Netflix movie ratings [19].

Finally, several papers study the footprint users leave online. For example, two studies show how many attributes users leave on different social networks and what is their consistency [12], [8].

8 Discussion

Matching entire social networks. Even though our correlation techniques scale to entire social networks, one might counter that *crawling* them proves infeasible, and hence large-scale attacks remain theoretical. We believe however that, with appropriate resources, collecting the necessary data is indeed realistic for two reasons. Firstly, one only needs to gather four attributes for each account: the username, real name, location and profile photo. Such information can often be obtained easily through APIs, and potential rate limits can be bypassed by proxying the requests through multiple computers (from a cloud for example). Secondly, some social networks such as Facebook and Twitter maintain user directories[5][6], listing usernames and real names of all their users for direct access, obliviating much of the need for recursive crawling.

Defenses. Although it takes some effort, the best defense against the type of correlation we discuss is straightforward: using different usernames, real names, locations and profile photos on each social network. An important aspect is however to consider *all* social networks, as correlation chains can link accounts even if all the features are different between a specific pair.

Extensions. Our correlations can achieve between 50% to 80% true positive rates for a false positive rate of zero depending on the social networks we consider. One could use the pairs of accounts that we link as a starting point and then apply further graph matching techniques to correlate the remainder, as done by Narayanan et al. [20].

Other applications. Not all applications of account correlation raise privacy concerns. For example, account similarity can help to detect fake accounts. Indeed, there are bots that use real names and photos scraped from social networks to create fake identities on other social networks to avoid detection. A specific advantage of using the Jaro distance and perceptual hashes for computing similarity (in contrast to more complex options) is their low resource demands, making them suitable for large-scale screening of newly created accounts in real-time.

9 Conclusions

Our work is the first study that sheds light on the ease of correlating accounts across different social networks by exploiting a combination of what users voluntarily publish in their profiles. Even though intuitively one may indeed anticipate the threat of such attacks, we consider the main contribution of our work in devising the actual correlation techniques; evaluating their performance on several major social networks; analyzing the root causes that enable their success; and finally demonstrating that these attacks prove feasible in practice even when matching across entire social networks.

Our results show in particular that when we combine several features extracted from user profiles attributes (usernames, real names, cross names, location, photo, and face) we reach excellent correlation performance, with improvements up to 100% over username alone. Specifically, we can match accounts between any pair-wise combination of Twitter, Facebook, Google+, and Flickr with a true positive rate of about 90% for

[5] https://twitter.com/i/directory/profiles/

[6] https://www.facebook.com/directory/

a false positive rate of 10^{-3}, and any combination between Myspace and one of these four with a true positive rate of about 60%. From our study, we also derive important insights on the matching power of different features and of their combinations that can be extrapolated to other features as well. Regarding the metrics, we found that, at very large scale, the most straightforward equality metric is essentially as good as more mismatch-tolerant similarity metrics.

We come to these numbers by using an extensive ground-truth set collected by crawling Google+ profiles. To explore how the results generalize to the overall user population, we also inspect a second ground-truth set derived from a large, independent list of email addresses. We find that while performance decreases, it remains high at 60% true positive rate for 10^{-3} false positives, meaning that we can correctly associate hundreds of thousands of accounts.

References

1. Bing Maps API. `http://www.microsoft.com/maps/developers/web.aspx`.
2. The CIA knows when you're hitting the gym. `http://www.itworld.com/it-management/341731/cia-knows-when-youre-gym`.
3. Opencv. `http://opencv.org`.
4. Phash. `http://www.phash.org`.
5. A. Acquisti, R. Gross, and F. Stutzman. Faces of facebook: Privacy in the age of augmented reality. In *BlackHat*, 2011.
6. M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *RAID*, 2010.
7. S. Bartunov, A. Korshunov, S.-T. Park, W. Ryu, and H. Lee. Joint link-attribute user identity resolution in online social networks. In *SNA-KDD Workshop*, 2012.
8. T. Chen, M. A. Kaafar, A. Friedman, and R. Boreli. Is more always merrier?: a deep dive into online social footprints. In *WOSN*, 2012.
9. W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, 2003.
10. T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, 2004.
11. T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. In *ICWSM*, 2011.
12. D. Irani, S. Webb, K. Li, and C. Pu. Large online social footprints–an emerging threat. In *SocialCom*, 2009.
13. J. Kerekes. Receiver operating characteristic curve confidence intervals and regions. *Geoscience and Remote Sensing Letters, IEEE*, 5(2):251 –255, april 2008.
14. M. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *MM*, 2010.
15. M. A. Mishari and G. Tsudik. Exploring linkability of user reviews. In *ESORICS*, 2012.
16. M. Motoyama and G. Varghese. I seek you: searching and matching individuals in social networks. In *WIDM*, 2009.
17. M. Nanavati, N. Taylor, W. Aiello, and A. Warfield. Herbert west: deanonymizer. In *HotSec*, 2011.
18. 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 *IEEE Symposium on Security and Privacy*, 2012.

19. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.
20. A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.
21. D. Perito, C. Castelluccia, M. Ali Kâafar, and P. Manils. How unique and traceable are usernames? In *PETS*, 2011.
22. Social Intelligence Corp. `http://www.socialintel.com/`.
23. M. Srivatsa and M. Hicks. Deanonymizing mobility traces: Using social network as a side-channel. In *CCS*, Oct. 2012.
24. L. Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine, and Ethics*, 25(2–3):98–110, 1997.
25. G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.
26. G.-w. You, S.-w. Hwang, Z. Nie, and J.-R. Wen. Socialsearch: enhancing entity search with social network matching. In *EDBT/ICDT*, 2011.

A Face similarity

Feature extraction for each user's profile image involves the following steps:

- Face and eye coordinate detection using OpenCV's pre-trained Haar cascade filters.
- Face size and scale normalization using the eye coordinates, such that each face becomes 64 pixels wide by 80 pixels high.
- For images where eye coordinate detection fails, substitute the average eye coordinate positions from images where detection succeeded.
- Performing a 2-dimensional Discrete Cosine Transform (DCT) on each image, and store the coefficients as features.

The stage for generating user similarity scores for users of a pair of social networks involves the following steps:

- Train a 16-mixture user-independent GMM on all image features for which the original eye coordinate detection succeeded, from one of the two social networks.
- Adapting the user-independent GMM to obtain a user-specific GMM for each user.
- Obtain similarity scores for user pairs by computing the log-likelihood of the features of images for the other social network, with the user-specific GMMs from the first social network.