6-5-2008

Linkability in Activity Inference Data Sets

Jeffrey Fielding
Dartmouth College

Linkability in Activity Inference Data Sets

Senior Honors Thesis

Jeffrey Fielding
Advisors: Tanzeem Choudhury and David Kotz

June 5, 2008

Dartmouth Computer Science Technical Report TR2008-623

Abstract

Activity inference is an active area of ubiquitous computing research. By training machine learning algorithms on data from sensors worn by volunteers, researchers hope to develop software that can interact more naturally with the user by inferring what the user is doing. In this thesis, we use the same sensor data to infer which volunteer is carrying the sensors. Such inference could be useful – for example, a mobile device might infer who is carrying it and adapt to that user's preferences. It also raises some privacy concerns, since an attacker could learn more about a user by linking together several sensor traces from the same user. We develop a model to differentiate users based on their sensor data, and examine its accuracy as well as the potential benefits and pitfalls.

1 Introduction

As anyone who has used a search engine knows, data mining has incredibly useful applications, not only for researchers and businesses, but also for the end user. Software that can respond intelligently to queries, recommend items based on the user's interests, or filter spam from his inbox is a boon to productivity as well as entertainment. As mobile devices become more powerful and pervasive, software that recognizes and responds to the user's context will become increasingly important. Activity inference algorithms can use sensors worn by the user to do just that.

To train these algorithms, however, it helps to have a large data set of sensor traces collected by individuals as they go about their daily activities. The same data sets could also be used to study real-world social networks [8], or gather health and fitness statistics. It may be advantageous, therefore, to make such data sets publicly available for researchers to explore. Before doing so, however, it is important to consider the privacy implications.

Even if the data set is anonymized, it may still be possible for an attacker to infer more than was intended. For example, Netflix released anonymized movie ratings from its customers as part of a competition to improve on its recommendation algorithm. Narayanan and Shmatikov showed that it was possible to link these "anonymous" ratings to users on IMDB [4].

In this thesis we analyze the linkability of a large set of sensor traces from 25 subjects: are an individual's activity patterns consistent enough, and sufficiently distinct from others' activity patterns, that we can reliably group traces by subject knowing only the activity patterns? If we consider more fine-grained features, how much better can we do? Furthermore, how can we quantify the ability of an adversary to infer information about the subjects?

In Section 2, we provide an overview of the data set and the theoretical background for our work. We describe our approach in Section 3, including our assumptions about the adversary and our clustering methods. We present our results in Section 4. In Section 5, we suggest areas for further study, including other inference attacks that link sensor traces as a first step.

2 Background

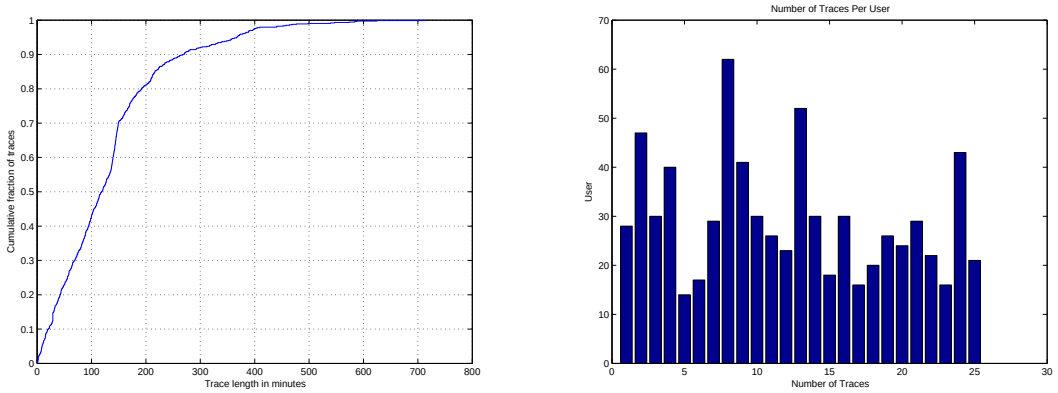
2.1 Data Set

We analyze a subset of the data collected for the University of Washington's Dynamic Social Network study. The study used the Mobile Sensing Platform (MSP), a wearable sensor pack and PDA designed to infer the activity of the individual wearing the device. The MSP includes a triaxial accelerometer, compass, visible and infrared light, temperature, barometric pressure, and humidity sensors. It also collects privacy-sensitive audio features from its microphone and the MAC addresses and signal strengths of WiFi access points picked up by the PDA [8]. This thesis does not consider the audio or WiFi data, however.

Our data set consists of 735 traces from 25 subjects over three weeks. In total, we analyzed 1,628 hours of sensor data. The average trace length was 133 minutes. Figure 1(a) shows the distribution of trace lengths. Figure 1(b) shows the distribution of traces by subject. Traces are continuous; however, some subjects collected multiple traces on the same day. We consider the effect of combining subjects' traces for each day in Section 4.3.

Often, raw sensor values at a particular instant are less informative than functions of the sensor values within some window. For example, the low-frequency FFT coefficients of the acceleration are far more useful in detecting a user's gait than the raw acceleration values. We use the same 651 features as Lester et al., including the mean, variance, and FFT coefficients, among other functions of the sensor values [3].

In addition, we trained an activity classifier on a smaller, labeled data set using the boosting algorithm described by Lester et al. [3]. Boosting combines many weak classifiers – in this case, simple thresholds on feature values – to



(a) Cumulative distribution of trace lengths.

(b) Number of traces per subject.

Figure 1: Distributions of traces by length and subject.

create a more accurate classifier by focusing on the most difficult training examples [6]. The classifier can identify nine activities: sitting, standing, walking, climbing (and descending) stairs, riding in an elevator (up or down), watching TV, and brushing teeth.

2.2 Information Theory

We approach the problem from an information-theoretic perspective. The most fundamental quantity we deal with is *entropy*. The entropy of a random variable X is defined as $H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$. Entropy is a measure of uncertainty; it can be thought of as the expected number of bits required to encode the outcome of a random variable knowing its probability distribution. For example, the entropy of a fair coin is 1 bit: since there are two equally likely outcomes, the outcomes can be optimally encoded as a single bit number.

Consider the owner of a trace to be a 25-outcome random variable. A naïve encoding of the outcome would simply be an integer between 1 and 25, which could be represented as a 5-bit number. However, 5 bits can actually encode 32 outcomes. In fact, one could compress the outcomes into $\log_2(25) \approx 4.6439$ bits per outcome. If one knew the distribution of outcomes, one could do better. Since some subjects have more traces in the data set than others, one could assign those subjects a shorter encoding. The entropy of the distribution of subjects in our data set, for example, is 4.5348.

Another quantity we make use of is the *Kullback-Leibler divergence*. The K-L divergence measures the divergence between two probability distributions. For discrete distributions P and Q, $D_{KL}(P \parallel Q) = \sum_i P(i) \log_2 \frac{P(i)}{Q(i)}$. Since the K-L divergence is not symmetric, we use $D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)$ as the distance between two probability distributions. $D_{KL}(P \parallel Q)$ can be thought of as the expected number of extra bits required to encode an outcome from P using an optimal encoding for Q. If Q and P are similar – that is, if Q is a good model for P – then the divergence will be small.

2.3 Anonymity, Pseudonymity, and Linkability

Privacy is a multifaceted concept for which there is a significant amount of terminology and numerous metrics. Pfitzmann and Hansen consolidate much of the terminology in the context of sending and receiving messages across a network [5]. Their definitions are helpful, although they do not always apply directly to our analysis.

Our data set is *pseudonymous*: traces are labeled with integer user identifiers, not real names. Nor do we have a mapping from the integer identifiers to real names or any other unique identifier for the subjects. While an adversary with significant background knowledge might be able to discover such a mapping, we do not attempt to do so. Instead,

we consider the problem of grouping *unlabeled* traces by subject, using only the features or inferred activities in each trace.

Anonymity refers to a subject being indistinguishable (from the adversary's perspective) from other subjects in an *anonymity set*. A simplistic measure of anonymity, therefore, is simply the size of the anonymity set. As Pfitzmann and Hansen note, however, anonymity also depends on the distribution of the *items of interest* (IOIs) – in our case, the traces. For example, if 24 subjects each had only one trace, and the remaining 711 traces belonged to the 25th subject, then it would be reasonable to assume that any given trace belongs to the 25th subject. To account for such non-uniform distributions, Serjantov and Danezis suggest using the entropy of the *anonymity probability distribution* – that is, the distribution of the probability that the item of interest belongs to each user [7]. If H is the entropy of the anonymity probability distribution, then the size of an equivalent anonymity set (assuming a uniform distribution over the subjects) is simply 2^H.

As mentioned in Section 2.2, the entropy of the distribution of subjects – that is, the anonymity using Serjantov and Danezis' measure – in our data set is 4.5348. We calculate this as $-\sum_{s \in S} \frac{N_s}{N} \log_2 \left(\frac{N_s}{N} \right)$ where S is the set of subjects, N_s is the number of traces for subject s, and N is the total number of traces. The size of an equivalent anonymity set assuming a uniform distribution is $2^{4.5348} \approx 23.18$, slightly smaller than the actual size of the anonymity set (25).

Unlinkability complements anonymity. Pfitzmann and Hansen define two or more items of interest as unlinkable if an attacker cannot determine whether they are related (e.g. belong to the same subject) or not. *Linkability* is the opposite: the degree to which an attacker can determine whether or not the IOIs are related. Entropy can be a useful measure of linkability as well. Clauß measures the unlinkability of two IOIs as $H(p_r, p_{\neg r})$, where p_r is the probability that the IOIs are related (and $p_{\neg r} = 1 - p_r$) [1].

We can use Clauß' measure to quantify the unlinkability of our data set. p_r is simply the probability of drawing from the data set (without replacement) two traces from the same subject: $p_r = \sum_{s \in S} \frac{N_s}{N} \cdot \frac{N_s - 1}{N - 1} = 0.0452$. Thus, the unlinkability of our data set is 0.2657. Compare this to a uniformly distributed set of traces from 25 users, in which p_r would be $\frac{1}{25} = 0.04$. Such a data set would have an unlinkability of $H\left(\frac{1}{25}, 1 - \frac{1}{25}\right) = 0.2423$. Note that the unlinkability of our data set is actually higher than that of a uniformly distributed data set, even though p_r is higher for our data set. This is because Clauß' unlinkability metric measures the attacker's degree of certainty that two traces are linked (if $p_r > 0.5$) or not linked (if $p_r < 0.5$).

3 Approach

3.1 Attacker Model

We assume the attacker has a set of traces from a known number of subjects, N, but does not have a mapping of traces to subjects. The attacker's primary goal is to to cluster the set of traces by subject as accurately as possible. We do not attempt to determine the real names of the subjects, but only seek to assign the traces to clusters isomorphic to the original pseudonyms. Even if the clusters are not perfectly accurate, we seek to reduce the anonymity of traces in each cluster, and to concentrate traces from the same user in as few clusters as possible.

To do this, we cluster traces based on the divergence of various distributions extracted from the traces. We first attempt to cluster using only activity distributions, then consider the distributions of the 651 features described in Section 2.1. We do not consider other details about the traces such as start or end times, trace length, or the order of the feature or activity values.

We consider the effect of two forms of background knowledge gathered from our data set, knowing the owner of each trace: the distribution of users, and the overall distribution of activities or feature values across all traces belonging to each user. We do not consider other forms of background knowledge, although we suggest some for future work in Section 5.

3.2 Clustering Methods

We use hierarchical clustering to group traces by their similarity to each other as measured by the K-L divergence of their activity or feature distributions. Hierarchical clustering builds a binary tree from the leaves (the items being clustered) up by repeatedly joining the closest clusters until only one cluster remains. We can then choose a threshold

distance to produce the desired number of clusters. The distance between clusters is a function of the distances between traces in the clusters. We tried using the minimum, maximum, and average distances, and found that the maximum worked best. [1]

We use two methods to evaluate the effect of background knowledge. We assume the attacker has a model of each user's activity or feature distribution. First, we simply assign each trace to the user with the closest model. Second, we use the distribution of users to compare models using Bayes' theorem: $P(s_i|F) = \frac{P(F|s_i) \cdot P(s_i)}{P(F)}$. Here, $P(s_i|F)$ is the conditional probability that the trace belongs to subject s_i given the feature or activity distribution F. $P(F|s_i)$ is the conditional probability of the feature or activity distribution given s_i, which we compute from the model for s_i. $P(s_i)$ is the prior probability that the trace belongs to s_i, which we compute from the distribution of traces among users. Finally, $P(F)$ is the prior probability of the feature distribution, although this is unnecessary for model comparison, since it is merely a scaling factor.

We represented activity distributions as histograms with one bin per activity. For feature values, we used both a normal distribution and a 100-bin histogram with bins evenly spaced within two standard deviations of the feature's mean value across all traces. We focus on the histogram representation, since feature values were rarely normally distributed. When using histograms, we added a pseudocount of 1 to each bin to account for rare values.

3.3 Evaluation Metrics

To evaluate the performance of our models, we used several metrics. The simplest metric assumes a one-to-one mapping of clusters to subjects. We choose the mapping that maximizes the accuracy (the percentage of traces that end up in the "correct" cluster). This accounts for the fact that the attacker's goal is to group traces by subject, not to recover the original pseudonyms, which are useless to the attacker since we assume the attacker has no data labeled with the original pseudonyms.

Often, however, clusters contain traces from multiple subjects. Even so, the distribution of subjects in a cluster will often have a lower entropy than the distribution of subjects across all traces. Thus, even though the subjects in a cluster are not easily distinguishable, we gain some information about the owner of a trace by knowing which cluster it belongs to. This information gain corresponds to a reduction of anonymity. For example, by reducing the entropy of a cluster by 1 bit, we effectively halve the size of the equivalent uniform anonymity set.

Note that small clusters will have low entropy simply because of their size. For example, the entropy of a cluster containing one trace will be 0, but so will the entropy of any randomly chosen one-element subset. Figure 2 shows the average entropy of a randomly chosen subset of the traces. We define the anonymity reduction of a cluster as the average entropy of a random subset of the same size minus the entropy of the cluster.

To gauge the overall reduction in anonymity, we define the expected anonymity of a trace after clustering to be the expected anonymity of the cluster containing a randomly selected trace. That is, the expected anonymity of a trace is the average cluster anonymity, weighted by cluster size.

Furthermore, a user's traces are often distributed across several clusters. To measure the unlinkability of a user's traces, we'll use the entropy of the distribution of the user's traces across clusters. Users whose traces are concentrated in a few clusters will have a lower unlinkability (higher linkability) than users whose traces are spread more evenly across many clusters. We compute the average of each user's linkability, weighted by the number of traces belonging to the user, to measure the overall linkability of traces after clustering. We do not use Clauß' measure of linkability in this case, since we do not have a good estimate of the probability distribution of the possible owners of a trace.

4 Results

We evaluated the performance of clustering under variations of the following:

- Clustering algorithm
- Background knowledge or lack thereof
- Activity or feature distribution
  - If using features, which one(s)

[1] We found that K-means clustering using a cosine distance performs similarly; however, we will focus on hierarchical clustering.

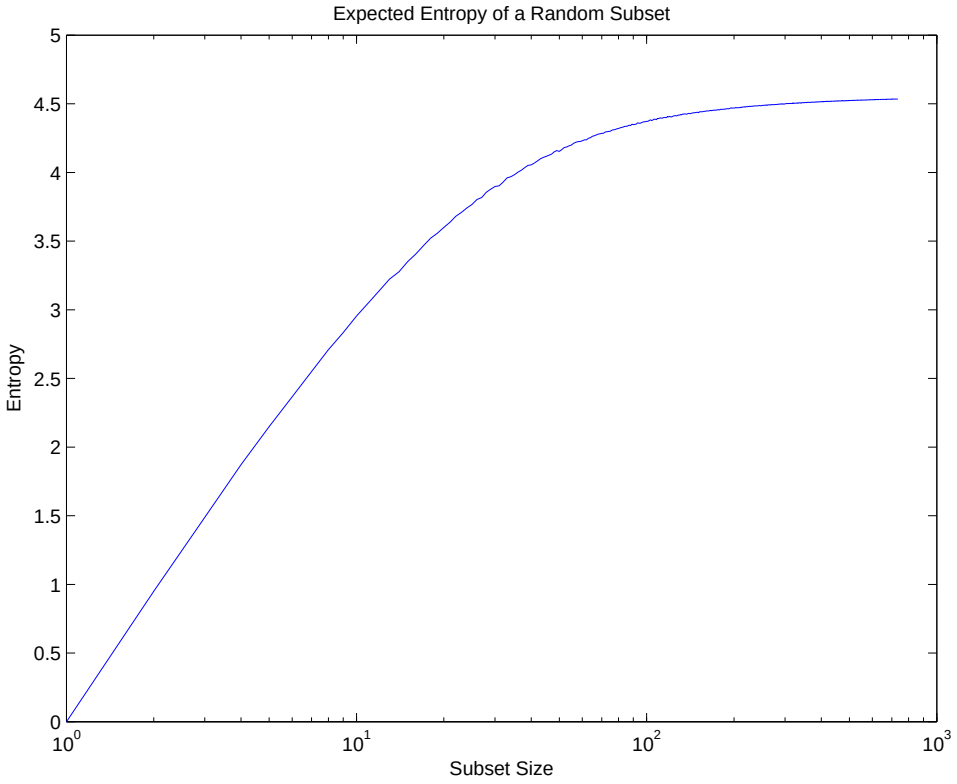
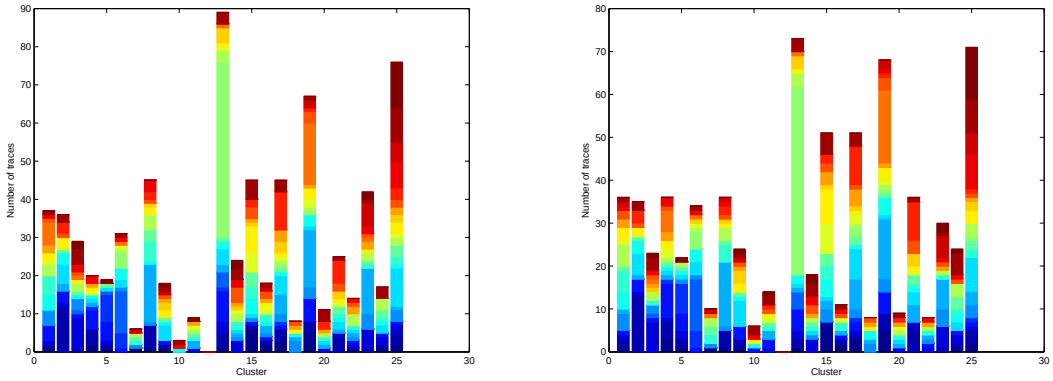


Figure 2: The average entropy of users in a randomly chosen subset of traces, by subset size.

- Whether or not multiple traces from the same day are combined

4.1 Using Background Knowledge: User Models

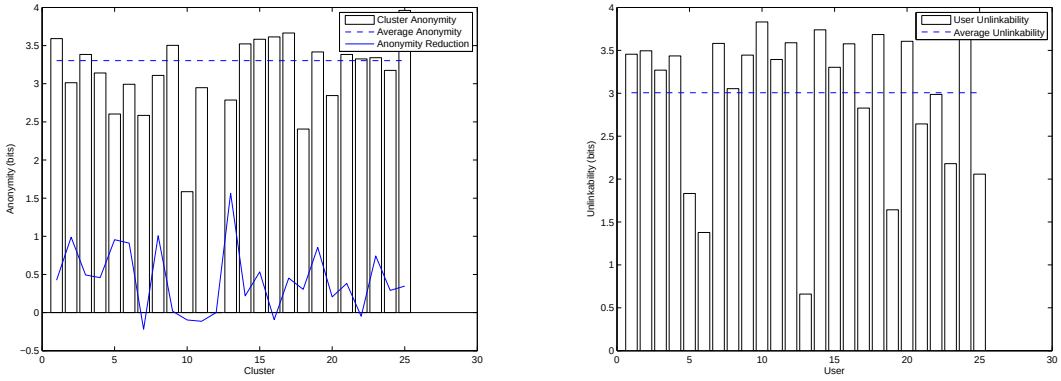
The premise of our clustering approach is that traces from the same user will tend to be more similar to each other than to traces from other users. We might expect a user's traces to approximate some typical distribution for the user. We considered the case that the attacker has as background knowledge a model of each user's feature or activity distributions. We created a model for each user by creating a histogram of the user's activities or feature values across all of the user's traces. We then assigned each trace to the closest model using either K-L divergence or Bayesian analysis.



(a) Clusters using K-L divergence of activity distributions.

(b) Clusters using Bayesian inference to compare activity distribution models.

Figure 3: Results of clustering on activity distributions by assigning each trace to the user with the closest model. Each color represents a different user.



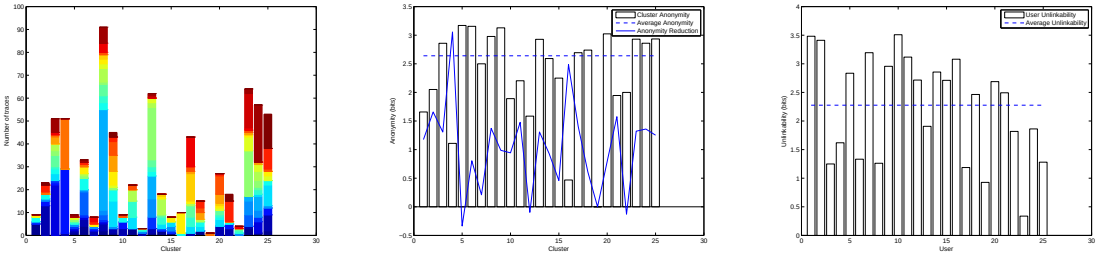
(a) Anonymity

(b) Linkability

Figure 4: Anonymity of clusters and linkability of users after assigning each trace to the user with the closest activity distribution model (as measured using K-L divergence).

As shown in Figure 3 and Table 1, Bayesian model comparison performs only slightly better than simply using the K-L divergence. Both get about 25% accuracy and reduce the average anonymity of a cluster by approximately 1.2 bits. Figure 4 shows the per-cluster anonymity and per-user unlinkability when comparing models using the K-L divergence. Overall, model comparison provided an improvement over the unclustered data set; however, the fact that

only 25% of traces are closest to the average activity distribution of the owner suggests that traces from the same user are only slightly more consistent than traces from different users. In fact, Figure 3 shows that user 12's cluster is empty: all of user 12's traces are closer to some other user's model than to his.



(a) Clusters (b) Anonymity (c) Linkability

Figure 5: Results of clustering using closest user model for "Accelerometer mag FFTBands 2 512s".

As we expected, using feature distributions significantly improves the performance of clustering over clustering using activity distributions. Table 2 shows the performance of the top ten features. The most accurate feature has an accuracy of 38% and reduces the average anonymity by 1.86 bits. Furthermore, users' traces are more linkable (by about 0.6 bits) when using the distribution of the top features than when using the distribution of activities. Note that all of the top 10 features are derived from the accelerometer, and most are FFT coefficients. This suggests that gait is most likely the most distinguishing feature between users that is also most consistent for each user.

Figure 5 shows the results of clustering using the "Accelerometer mag FFTBands 2 512s" feature – that is, the second band of the FFT of the magnitude of acceleration over a 512 second window. This is actually the second most accurate feature using this clustering method, but it is the most consistently accurate across clustering methods, so we present it for comparison.

4.2 Clustering With No Background Information

Normally, the attacker would not have a model for the users in the data set. We therefore focused most of our analysis on clustering traces based only on their similarity to each other.

Table 3 shows the performance of clustering on activity distributions using various clustering methods. Of the hierarchical clustering methods, we found it works best to define the distance between clusters as the maximum distance between any pair of elements from the two clusters; unless stated otherwise, all hierarchical clustering experiments use this metric. For comparison, we also tried K-means clustering, using the activity distributions as 9-dimensional vectors. The "city block" or L_1 distance is the sum of the absolute differences of the coordinates; the "cosine" distance is $1 - \cos(\theta)$, where θ is the angle between the vectors. We found that hierarchical clustering generally performs about as well or better than K-means using the best distance metric, so we focus on hierarchical clustering.

Figure 6 shows the dendrogram of the traces, clustered hierarchically by activity distribution. Figure 7 shows the resulting clusters and the effect on the anonymity and linkability of the traces.

Table 4 shows the performance of the top ten features using hierarchical clustering. Again, all of the top ten features are derived from the accelerometer.

Table 1: Performance of clustering using an activity histogram model.

Method	Accuracy	Expected Anonymity	Unlinkability
No Clustering	N/A	4.53	N/A
Bayesian Inference	25.6%	3.29	3.04
Closest overall distribution	24.9%	3.30	3.01

Table 2: Performance of clustering using the K-L divergence of feature distributions between a trace and user models.

Feature	Accuracy	Expected Anonymity	Unlinkability
Accelerometer Z FFTBands 8 512s	38.0%	2.67	2.43
Accelerometer mag FFTBands 2 512s	37.3%	2.64	2.28
Accelerometer X Mean 512s	36.5%	2.92	2.48
Accelerometer mag FFTBands 1 512s	34.6%	2.68	2.34
Accelerometer mag FFTBands 7 512s	34.1%	3.00	2.67
Accelerometer mag FFTBands 8 512s	33.9%	2.86	2.59
Accelerometer Z FFTBands 7 512s	33.8%	2.98	2.62
Accelerometer mag Mean 512s	33.8%	2.69	2.35
Accelerometer Y FFTBands 7 512s	32.8%	2.94	2.61
Accelerometer X FFTBands 1 512s	32.7%	3.03	2.62

Table 3: The performance of clustering on activity distributions using various clustering methods.

Method	Accuracy	Expected Anonymity	Unlinkability
No Clustering	N/A	4.53	N/A
Hierarchical, minimum distance	11.7%	4.25	0.30
Hierarchical, average distance	18.4%	3.82	1.18
Hierarchical, maximum distance	21.0%	3.46	2.66
K-means (square Euclidean)	16.1%	3.67	3.33
K-means (city block)	16.2%	3.55	3.50
K-means (cosine)	21.3%	3.38	3.02

Table 4: Performance of hierarchical clustering using the K-L divergence of feature distributions.

Feature	Accuracy	Expected Anonymity	Unlinkability
Accelerometer mag FFTBands 2 512s	31.3%	2.87	1.97
Accelerometer Trapz Intg 5 5sec	30.7%	3.00	1.94
Accelerometer Trapz Intg 5 30sec	30.1%	2.92	2.05
Accelerometer mag FFTBands 1 512s	29.3%	2.94	1.92
Accelerometer Z FFTBands 8 512s	28.3%	2.90	1.97
Accelerometer Trapz Intg 5 60sec	27.9%	2.99	1.98
Accelerometer Trapz Intg 5 15sec	27.5%	2.95	2.14
Accelerometer mag Mean 512s	27.0%	3.05	1.91
Accelerometer Y FFTBands 8 512s	26.7%	3.20	1.99
Accelerometer Z FFT Band 2 to 3 512s	26.3%	3.22	2.69

Table 5: The performance of clustering after combining traces from the same user on the same day.

Method	Feature	Accuracy	Expected Anonymity	Unlinkability
No Clustering		N/A	4.62	N/A
Hierarchical	Activity	26.5%	2.86	2.35
Hierarchical	Accelerometer mag FFTBands 2 512s	34.5%	2.43	1.57
Closest model	Activity	34.5%	2.60	2.40
Closest model	Accelerometer mag FFTBands 2 512s	48.9%	2.00	1.71

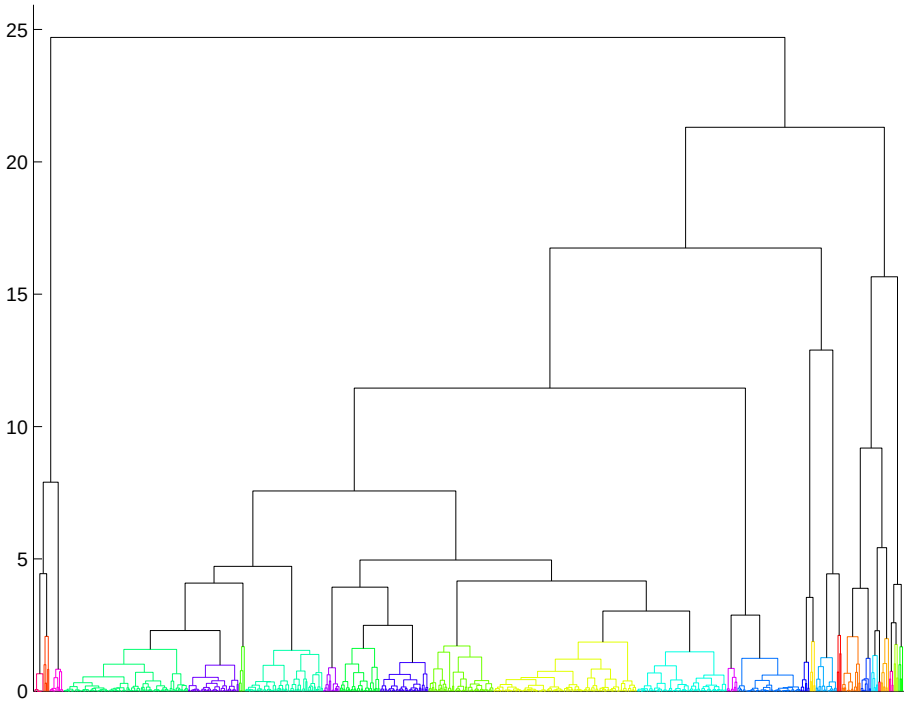
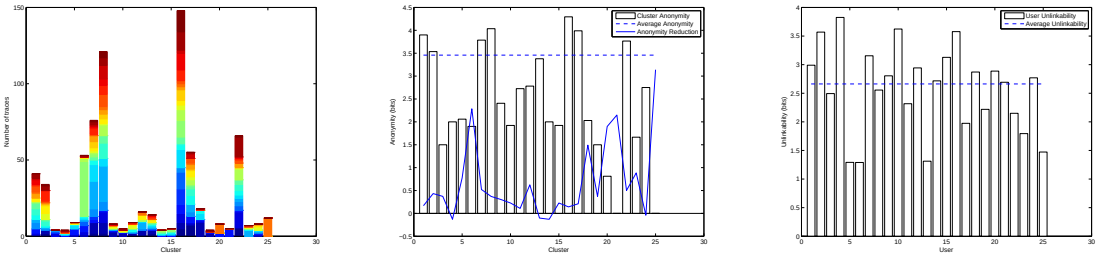
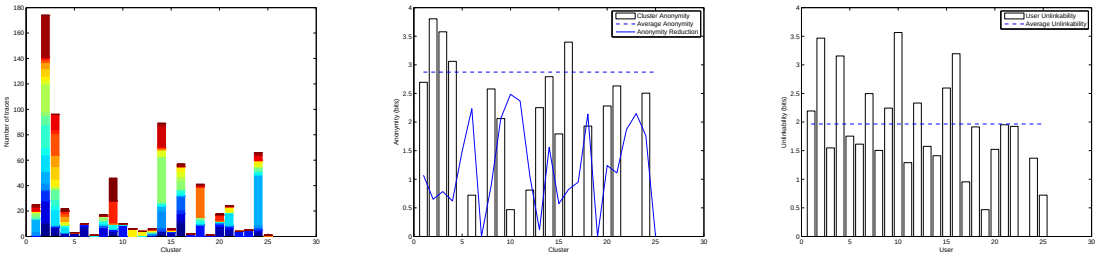


Figure 6: Hierarchical clustering of traces using the symmetric K-L divergence of their activity distributions. Distances between clusters are calculated as the maximum distance between traces in the clusters. The colored sub-trees at the bottom of the graph represent the 25 clusters. The height of each horizontal line is the distance between the clusters it connects.



(a) Clusters (b) Anonymity (c) Linkability

Figure 7: Results of hierarchical clustering on activity distributions.



(a) Clusters (b) Anonymity (c) Linkability

Figure 8: Results of hierarchical clustering using "Accelerometer mag FFTBands 2 512s".

4.3 Effect of Longer Traces

Table 5 shows a significant improvement in all the clustering methods we've discussed so far after combining traces from the same user on the same day. This reduces the number of (combined) traces to 313, although the set of combined traces actually has a slightly higher anonymity (4.62) since the traces are more evenly distributed among users.

4.4 Random Clustering

For comparison, we generated 10,000 random assignments of traces to clusters consistent with the distribution of traces among users and computed our evaluation metrics on these. That is, the distribution of traces among clusters was the same as the distribution of traces among users. The average accuracy was 11%, while the average linkability and anonymity were both 3.89.

4.5 Analysis

We've shown that an adversary can use clustering to group traces such that the anonymity of traces with each cluster is less than the anonymity of the traces in the unclustered data set. All of our clustering methods performed significantly better than the average randomly generated set of clusters, despite the fact that only our Bayesian clustering method took the distribution of traces into account, while our randomly generated clusters followed the original distribution of traces exactly.

As we expected, clustering using feature values – particularly accelerometer features – performed significantly better than clustering on activity distributions. Having a model of each user allowed us to classify traces more accurately; interestingly, however, these model-based methods tended to produce clusters with significantly higher unlinkability than hierarchical clustering using the same distributions.

5 Future Work

5.1 Improved Metrics

While our clustering methods provide the adversary with some information, they do not estimate the probability that a trace belongs to a user. This causes two major problems:

First, without an estimate of the probability distribution it impossible to quantify the anonymity or linkability of the clusters. We used the actual owners of the traces to compute the anonymity and linkability of the clusters. In a real-world scenario, the adversary could be reasonably sure that clustering would reduce the anonymity and increase the linkability of the traces, but would have no way to quantify it.

Second, it's difficult to link traces across clusters. For example, while a linkability of 2 implies that on average, a user's traces are spread across about four clusters, it's not obvious how to estimate which clusters those are.

Clustering might serve as a starting point for other algorithms that could provide better probability estimates and optimize the clusters accordingly.

5.2 Location Attacks

It may be possible to estimate the relative location of users from their sensor traces. A simple model might use the compass and accelerometer, for example. The WiFi readings could also be used to estimate the user's location. Such a (relative) location trace could be useful both to cluster users and possibly to reveal the actual identity of a user (given enough background information).

A user might tend to follow certain location patterns based on the pathways in the buildings she typically visits. It would also be possible to infer where along those paths the user tends to spend the most time (e.g. in her office). Traces with similar location patterns would be more likely to belong to the same user.

On the other hand, if the adversary has already linked the traces, he might be able to discover the user's identity based on where she spends the most time. Krumm demonstrated location attacks against GPS traces [2], but similar attacks might be possible against less accurate traces – perhaps even relative location traces, given enough background information. For example, one could compare the relative location traces in our data set against common pathways at the University of Washington.

6 Conclusion

This paper shows that an adversary can use simple clustering methods on the activity or feature distributions of traces in a data set to reduce the anonymity and unlinkability of the traces. If the adversary can tolerate significant misclassification errors, he can assign a pseudonym to each cluster and (under some isomorphism between the attacker's pseudonyms and the subjects in the data set) expect to correctly link around 30% of the traces. While this may not seem like a significant threat to privacy, we only had time to consider relatively simple methods. More sophisticated methods likely could achieve significantly better results.

References

[1] Sebastian Clauß. A framework for quantification of linkability within a privacy-enhancing identity management system. In *Emerging Trends in Information and Communication Security*, pages 191–205. Springer, 2006.

[2] John Krumm. Inference attacks on location tracks. In *Pervasive 2007*, pages 127–143. Springer, 2007.

[3] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative-generative approach for modeling human activities. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 766–772, Edinburgh, Scotland, July 2005.

[4] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset, November 2007.

[5] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology. Website, February 2008. Version 0.31 at http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf.

[6] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[7] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies*, volume 2482 of *Lecture Notes in Computer Science*, pages 259–263, 2003.

[8] D. Wyatt, T. Choudhury, and H. Kautz. Capturing spontaneous conversation and social dynamics: A privacy-sensitive data collection effort. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007.*, 4:213–216, April 2007.