Partial Device Fingerprints

Ciere, Michael; Gañán, Carlos; van Eeten, Michel

DOI
10.1007/978-3-319-71246-8_14

Publication date
2017

Document Version
Accepted author manuscript

Published in
proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases

Citation (APA)
Ciere, M., Gañán, C., & van Eeten, M. (2017). Partial Device Fingerprints. In *proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 222-237). (Lecture Notes in Computer Science; Vol. 10535). Springer. https://doi.org/10.1007/978-3-319-71246-8_14

Partial device fingerprints

Michael Ciere, Carlos Gañán, and Michel van Eeten

Delft University of Technology
m.ciere@tudelft.nl

Abstract. In computing, remote devices may be identified by means of *device fingerprinting*, which works by collecting a myriad of client-side attributes such as the device's browser and operating system version, installed plugins, screen resolution, hardware artifacts, Wi-Fi settings, and anything else available to the server, and then merging these attributes into uniquely identifying fingerprints. This technique is used in practice to present personalized content to repeat website visitors, detect fraudulent users, and stop masquerading attacks on local networks. However, device fingerprints are seldom uniquely identifying. They are better viewed as *partial device fingerprints*, which do have some discriminatory power but not enough to uniquely identify users. How can we infer from partial fingerprints whether different observations belong to the same device? We present a mathematical formulation of this problem that enables probabilistic inference of the correspondence of observations. We set out to estimate a correspondence probability for every pair of observations that reflects the plausibility that they are made by the same user. By extending probabilistic data association techniques previously used in object tracking, traffic surveillance and citation matching, we develop a general-purpose probabilistic method for estimating correspondence probabilities with partial fingerprints. Our approach exploits the natural variation in fingerprints and allows for use of situation-specific knowledge through the specification of a generative probability model. Experiments with a real-world dataset show that our approach gives calibrated correspondence probabilities. Moreover, we demonstrate that improved results can be obtained by combining device fingerprints with behavioral models.

1 Introduction

In networking, remote computers may be partially identified from information they disclose about themselves. This is called device fingerprinting. In the most general terms, device fingerprinting refers to any active or passive collection of meta-data for the purpose of host identification. These meta-data can for instance be browser user-agents, hard drive serial numbers, hardware artifacts such as clock skew, or implementation and configuration details of various protocols [10, 14, 19, 4, 22, 21]. Device fingerprints can be used both to distinguish hosts — for instance as a security mechanism against masquerading attacks on local networks — and to identify or track them, for instance to present personalized advertisements or to prevent fraud by recognizing blacklisted devices.

Ideally, the device fingerprints are both highly diverse and stable over time, like human fingerprints, so that they allow us to safely conclude whether or not two hosts are one and the same, even when considerable time has passed between observing them. In practice, however, device fingerprints are typically not completely unique across devices, which means they sometimes fail to distinguish hosts, or they are not stable, diminishing our ability to identify previously seen hosts over time. In some cases there is, in fact, a trade-off between uniqueness and stability, in that fingerprints can be made more unique by including unstable attributes.

Various scholars have addressed a lack of stability with heuristic methods that merge fingerprints based on some measure of similarity [10, 28]. However, little has been said about what to do when fingerprints are not completely unique. In this paper we consider *partial device fingerprints*. To be precise, the kind of fingerprint we are interested in is that which is not perfectly unique — multiple devices may have the same fingerprint — but it is stable and free of noise, at least in the time frame in which they are used, so we can distinguish devices with different partial fingerprints with absolute certainty. The challenge is to determine whether devices that show up with the same partial fingerprint are in fact one and the same.

If fingerprints are almost unique, we may be satisfied to map every fingerprint to a single device and accept the small number of false positive matches. As the diversity of fingerprints decreases, however, this may result in too many devices getting clumped together. A particularly unfortunate scenario is when the majority of fingerprints is unique, but a small number of fingerprints is shared by a disproportionate group of devices. In that case the discriminatory power of the unique fingerprints is diluted by the common ones.

So far, no-one has investigated automated methods for identifying devices from partial fingerprints. Our contribution is the development of a general-purpose probabilistic method that allows one to calculate for every pair of observations (o_1, o_2) with matching device fingerprints a *correspondence probability* $\mathrm{P}(\mathrm{device}(o_1) = \mathrm{device}(o_2))$, which reflects the plausibility that the observations originate from the same device. These correspondence probabilities have a self-contained measure of uncertainty, which may be large or small depending on the prevalence of a fingerprint. The strength of our method lies in the ability to use partial device fingerprints in combination with user modeling, which we demonstrate on a real-world dataset. Our mathematical formulation of the partial fingerprint problem reveals similarities to *data association* problems studied in Artificial Intelligence, and we draw upon methods previously developed for those problems. Our main developments in this regard are a new approach to dealing with an unknown number of entities and a way to exploit the variation in fingerprint prevalence.

2 Problem setting

The problem we are interested in can be formulated as follows. Let $O = \{o_1, o_2, ...\}$ be a set of observations, each one tagged with a partial fingerprint $f_1, f_2, ...$. The observations can take any form. Our only assumption is that they are exchangeable, which means O is invariant to permutations of the labeling. Let the unobserved *assignment* ω be a partition of $\{1, ..., |O|\}$ such that each subset in the partition contains the indices of observations made by a single device. The assignment defines an equivalence relation $o_a \sim o_b$ for observation pairs (o_a, o_b) belonging to the same device. We assume that fingerprints are *stable*; that is, if $f_a \neq f_b$ then $o_a \not\sim o_b$. The converse is not true: $f_a = f_b$ does not imply $o_a \sim o_b$, as different devices may have the same fingerprint.

Let the *complete-data* $Y = \{O, \omega\}$ be the union of the observed data O and the unobserved assignment ω. We only observe Y completely if the fingerprints are unique, because then we can infer ω by putting each fingerprint in its own equivalence class. If however we have partial fingerprints, then multiple assignments $\omega \in \Omega$ are plausible. Hence, our observations O can be seen as incomplete-data, and the assignment ω as missing data.

Analogous problems have been studied in the Artificial Intelligence field where they are referred to as *data association* problems [2]. A general formulation was given in [13]. Applications include object tracking, robotic map-building, surveillance, and citation matching [27, 15, 6, 18, 25, 3]. These problems have in common that observed objects or entities are to some degree indistinguishable, and the challenge is to determine whether two identical-looking objects are in fact one and the same.

In all data association problems, uncertainty about the correct assignment ω is unavoidable. This limits the usefulness of heuristic imputation of correspondences. Several AI scholars have applied probabilistic methods that capture the uncertainty due to measurement error and unpredictable trajectories in a generative probability model (e.g., [25, 7, 17]). This model can be designed to include prior information and assumptions about the appearance and behavior of entities.

The complete-data $Y = \{O, \omega\}$ can be written as a set of observation sequences $\{T_1, T_2, ..., T_N\}$, $N = |\omega|$, each one belonging to a unique device. Borrowing a term from the object tracking literature, we call such sequences *trajectories*. A probabilistic model $P(Y) = P(T_1, ..., T_N)$ can be designed to capture patterns and regularities in these trajectories. Using Bayes' law, this model then implies a conditional probability distribution over assignments given the observed data

$$P(\omega|O) = \frac{P(O, \omega)}{P(O)} \propto P(Y)$$

where we ignore the normalization constant $P(O)$ since it is constrained to make the probabilities sum to one. A correspondence probability $P(o_a \sim o_b)$ can then be estimated by summing over the unobserved assignment:

$$P(o_a \sim o_b) = \sum_{\omega \in \Omega:\ o_a \sim o_b} P(\omega|O) \propto \sum_{\omega \in \Omega:\ o_a \sim o_b} P(\omega, O)\,. \tag{1}$$

In the following section we explain how such a generative probability model can be set up, and in Section 4 we explain how correspondence probabilities can be computed according to (1).

3 Designing a complete-data probability model

We consider probability models that consist of three independent components: a prior distribution on the number of users, a prior distribution on the fingerprint counts, and a trajectory model. The parameters of these models can often be estimated from other data or prior information; for instance, if partial fingerprints are used to track website visitors who have cookies turned off, the complete-data model can be fitted to the data from users who have cookies turned on. Alternatively, model parameters can be estimated from the incomplete-data using a stochastic EM scheme that we will explain in Section 4.2.

3.1 Dealing with an unknown population size

The number of devices N observed in O is often unknown, and moreover it may vary over time as new observations come in. A similar situation was encountered in [16], who suggested estimating N by doing a grid search to optimize a pre-specified criterion. In the partial fingerprint context this approach is not attractive, as taking N to be a fixed parameter does not lead to a generative model. Furthermore, if one is looking to simultaneously estimate the size of multiple subpopulations, a grid search is not workable because of the curse of dimensionality.

An idea borrowed from ecology is to introduce an artificial supercommunity of size $S > N$ from which devices are randomly selected (e.g. [9]). To be precise, we fix S at some value that we know is larger than N, and for each device $i = 1, ..., S$ we introduce a random variable $z_i \in \{0, 1\}$ that determines whether device i is *available*. If a device has at least one observation in O, it is by definition available; otherwise it is either unavailable or available and unobserved. Hence, the size of the observed population is $N = \sum_{i=1}^{S} z_i$, which is a random variable.

One advantage of this approach is that we may introduce user-level parameters ν_i for $i = 1, ..., S$, which is of fixed size even though N is unknown. A second advantage is that it is straightforward to use prior information on N. In the simplest case we may set $N \sim U(1, S)$, but another option is to use $N \sim \text{Binomial}(S, \pi)$ with some informative prior on π that reflects prior knowledge. In addition, if we want the model to work for varying observation windows, we may choose to model the arrival and departure of users. This allows for online deployment of the model with a continuously expanding observation window.

3.2 Modeling variation in fingerprint prevalence

The overall prevalence of a partial fingerprint can be used to inform correspondence probabilities. The intuition is that if we have never seen fingerprint f

before and then observe it twice, it seems likely that these observations correspond to the same device; if on the other hand this fingerprint is known to occur frequently, we are less certain of this correspondence.

The simplest way to incorporate this idea in a probability model is to assume that fingerprints are drawn independently from a categorical distribution $f \sim \text{Categorical}(\mathbf{p})$ with $\mathbf{p} = (p_1, ..., p_K)$, $\sum_{f=1}^{K} p_f = 1$. Then for N users, the probability of generating fingerprint f_1 for the first user, f_2 for the second user, and so on, is

$$P(\mathbf{f}) = p_{f_1} p_{f_2} \cdots p_{f_N} = \prod_{f=1}^{K} p_f^{n_f} \,, \tag{2}$$

where n_f is the number of users with fingerprint f.

Now imagine two complete-data realizations Y_1 and Y_2, which can both be obtained by adding an observation o_{new} with a fingerprint f to Y_0, with one difference: in Y_1, the new observation is assigned to a device that exists in Y_0, while in Y_2, it is assigned to a new device. If Y_1 has N devices with n_f occurrences of fingerprint f, then Y_2 has $N+1$ devices with n_f+1 occurrences of fingerprint f. Hence, in the probability ratio $\frac{P(Y_1)}{P(Y_2)}$ all terms in (2) cancel out except for a $\frac{1}{p_f}$. This term is inversely proportional to p_f, the population frequency of fingerprint f, and consequently the hypothesis that o_{new} is made by a device previously seen with the same fingerprint is more plausible if the fingerprint is known to occur rarely.

In practice we do not know the true fingerprint population proportions $\mathbf{p}$. If we put a Dirichlet(α) prior on $\mathbf{p}$ for some $\alpha > 0$ and then integrate out the uncertainty in the fingerprint probabilities $\mathbf{p}$, the fingerprint counts follow a compound Dirichlet-Categorical distribution, which we can derive as

$$\begin{aligned} P(\mathbf{f}|\boldsymbol{\alpha}) &= \int_{\mathbf{p}} P(\mathbf{f}|\mathbf{p}) P(\mathbf{p}|\boldsymbol{\alpha}) \, \mathrm{d}\mathbf{p} \\ &= \frac{\Gamma(A)}{\Gamma(\alpha)^K} \frac{\prod_{f=1}^{K} \Gamma(n_f + \alpha)}{\Gamma(N + A)} \,. \end{aligned}$$

Now the probability ratio $\frac{P(Y_1)}{P(Y_2)}$ contains a term

$$\frac{N + A}{n_f + \alpha} \,. \tag{3}$$

This is inversely proportional to the smoothed fingerprint count $n_f + \alpha$, which is in line with our intuition: if we have seen few occurrences of f, we are inclined to believe that o_{new} belongs to a previously seen device.

Thus far we have assumed that the number of distinct fingerprints K that could ever occur is a finite number known to us. We may circumvent this by specifying a fingerprint distribution with an infinite number of fingerprints. If we fix A and write $\alpha = A/K$, then as K goes to infinity the term (3) goes to $(N + A)/n_f$. The corresponding generative process is known as the *Chinese* *restaurant process* (CRP) [1, 26]. Ignoring the order of users within groups, it has probability mass function

$$P(\mathbf{f}) = \frac{\Gamma(A)\, A^{|F|}}{\Gamma(A+N)} \prod_{f \in F} \Gamma(n_f) \tag{4}$$

where F is the set of observed fingerprints. The same distribution was introduced in the context of population genetics by Ewens [11].

A simple generalization of this distribution uses a discount parameter $0 < d < 1$ with

$$P(\mathbf{f}) = \frac{\Gamma(A)}{\Gamma(A+N)} \frac{d^{|F|}\Gamma(A/d + |F|)}{A/d} \prod_{f \in F} \frac{\Gamma(n_f - d)}{\Gamma(1-d)} \,. \tag{5}$$

This is sometimes called the two-parameter Poisson-Dirichlet distribution, and it is regularly used in similar situations such as the modeling of word frequencies.

3.3 Modeling user appearance and behavior

The last component of the generative probability model is a trajectory model $P(T)$. We assume that trajectories are independent, i.e. $P(T_1, ..., T_N) = \prod_{i=1}^{N} P(T_i)$, although the $P(T_i)$ may depend on shared hyperparameters.

We propose a simple trajectory model that can be used in all partial fingerprint situations. Assume that the number of observations in a trajectory follows a distribution $P(n = |T_i|)$ and that given $|T_i|$, the times at which these observations occur are drawn uniformly from the observation window. Then the probability of a trajectory T_i is

$$P(T_i) = P(|T_i|) \cdot |T_i|!$$

where the factorial term arises from the fact that the observations in a trajectory are always ordered in time.

More advanced trajectory models use domain-specific assumptions about the *coherence* of observation sequences. This can be achieved by modeling a sequence $T_i = o_1^i, o_2^i, ..., o_{|T_i|}^i$ as a Markov process

$$P(T_i) = p(o_1^i) \prod_{j=2}^{|T_i|} p(o_j^i | o_{j-1}^i)$$

where the transition probability $p(o_j^i | o_{j-1}^i)$ may use any information available at the observation-level.

If devices are observed repeatedly, there may be value in modeling the behavioral characteristics of users. For instance, a user may have a tendency to appear at a specific time of day. If this user's fingerprint appears at a non-typical time, we may be inclined to say this is a different user with the same fingerprint.

Such patterns can be used to inform correspondence probabilities by introducing user-level parameters ν_i in a trajectory model $P(T_i) = P(T|\nu_i)$. Alternatively, one may treat the user-level parameters as nuisance parameters and integrate them out of the likelihood. The probability of a trajectory T then changes from $P(T|\nu)$ to $P(T|\phi) = \int_\nu P(T|\nu)P(\nu|\phi)\,\mathrm{d}\nu$ where $P(\nu|\phi)$ is a hierarchical distribution. For convenient modeling choices this integral may have a closed-form solution.

3.4 Putting the components together

If we put these three model components together, we get a generative complete-data probability model

$$P(Y) = p(N) \cdot N! \cdot \frac{\Gamma(A)\, A^{|F|}}{\Gamma(A+N)} \prod_{f \in F} \Gamma(n_f) \cdot \prod_{i=1}^{N} P(T_i)\,. \tag{6}$$

The $N!$ term is needed to account for the fact that there is no natural order of users in our complete-data formulation.

4 Inference

4.1 Calculating correspondence probabilities from a complete-data probability model

The calculation of correspondence probabilities from a complete-data model as per (1) involves a summation over all possible assignments $\omega \in \Omega$. In general, to estimate any function of the complete-data $f(Y)$ conditional on the observations O and the probability model $P(Y)$ we must average over the uncertainty in ω. This is cumbersome. If there are fingerprints with more than a few observations, then a sum over $\omega \in \Omega$ is intractable, as the number of possible assignments $\omega \in \Omega$ grows combinatorially in the number of observations. A solution used extensively in the data association literature is to draw a sample $\omega^{(1)}, ..., \omega^{(M)}$ from $P(\omega|O) \propto P(Y)$ using a Markov-Chain Monte Carlo (MCMC) approach [12].

MCMC methods work by starting with a random guess ω_0 and then generating a Markov chain $\omega_0, \omega_1, \omega_2, ...$ using a transition kernel density $P(\omega'|\omega)$. For a suitably chosen transition density, the sequence $\omega_1, \omega_2, ...$ has the desired distribution as its stationary distribution. One way to achieve this is offered by the Metropolis-Hastings algorithm, which works as follows. First, the assignment is initialized to some value $\omega^{(0)} \in \Omega$. Then, in every iteration $t = 1, ..., T$, a candidate transition $\omega \to \omega'$ is randomly drawn from a proposal distribution $q(\omega'|\omega)$ and accepted with probability

$$A(\omega'|\omega) = \min\left(1, \frac{P(\omega'|O)\; q(\omega|\omega')}{P(\omega|O)\; q(\omega'|\omega)}\right)\,. \tag{7}$$

If the proposal is accepted we set $\omega^{(t+1)} = \omega'$, and otherwise $\omega^{(t+1)} = \omega^{(t)}$. If the proposal distribution is reversible, that is, $q(\omega|\omega') > 0$ if and only if $q(\omega'|\omega) > 0$, then the resulting Markov Chain is ergodic with stationary distribution $P(\omega|O)$, as desired.

The idea is to construct a proposal distribution that proposes small changes to the assignment, by at most a few correspondences. Fig. 1 shows four types of simple reversible transitions. It can easily be verified that every possible assignment can eventually be reached by making only such transitions. Various proposal mechanisms using some combination of the above transitions have been suggested in the data association literature [24, 23, 18, 25].

The computational efficiency of a Metropolis-Hastings algorithm designed in this manner lies in the fact that the acceptance probability (7) is always simple to compute, because most terms cancel out. The first half of the probability ratio in (7) can be written as

$$\frac{P(\omega'|O)}{P(\omega|O)} = \frac{P(\omega',O)}{P(\omega,O)} = \frac{P(Y')}{P(Y)} \,.$$

Plugging in the three-component model (6) yields

$$\frac{P(Y')}{P(Y)} = \frac{p(N') \cdot N'! \cdot \frac{\Gamma(A)\,A^{|F|}}{\Gamma(A+N')} \prod_{f\in F} \Gamma(n'_f) \cdot \prod_{i=1}^{N} P(T'_i)}{p(N) \cdot N! \cdot \frac{\Gamma(A)\,A^{|F|}}{\Gamma(A+N)} \prod_{f\in F} \Gamma(n_f) \cdot \prod_{i=1}^{N} P(T_i)} \,. \tag{8}$$

When the proposal is a simple transition shown in Fig. 1, Y and Y' differ by at most three trajectories T_i and only one fingerprint count n_f, which means almost all terms in (8) cancel out and a simple expression remains.

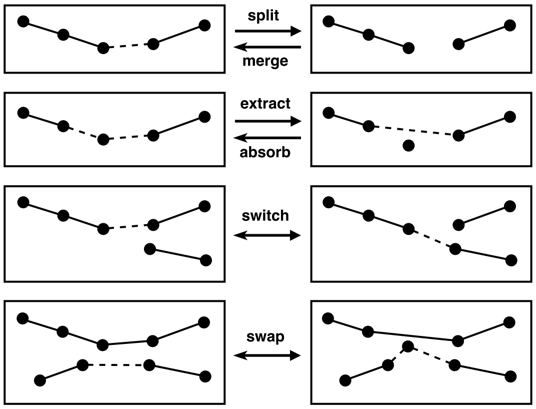

Fig. 1. Visualization of Markov transitions used in MCMC sampling from the conditional distribution of assignments

4.2 Parameter estimation

We may estimate the parameter vector $\boldsymbol{\theta}$ of the probability model by maximizing the marginal likelihood, obtained by summing out the unobserved assignment ω

$$L(\theta|O) = P(O|\boldsymbol{\theta}) = \sum_{\omega \in \Omega} P(O, \omega|\boldsymbol{\theta})\,.$$

The summation is intractable. A solution proposed by [25] is to use a stochastic Expection-Maximization (EM) scheme. The EM algorithm works by first setting the parameter vector to an initial guess θ_0 and then alternating the following two steps [8, 20]:

E: Compute the posterior distribution over the missing data ω given the observed data O and the current parameter guess:

$$\tilde{P}^{(t)}(\omega) = P(\omega|O, \boldsymbol{\theta}^{(t-1)})\,.$$

M: Update the parameters to $\theta^{(t)}$ by maximizing the expectation of the complete data log-likelihood with respect to $\tilde{P}^{(t)}$:

$$\begin{aligned} \theta^{(t)} &= \arg\max_{\boldsymbol{\theta}} \mathbf{E}_{\tilde{P}^{(t)}} [\log P(\omega, O|\boldsymbol{\theta})] \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{\omega \in \Omega} \left[\tilde{P}^{(t)}(\omega) \cdot \log P(O, \omega|\boldsymbol{\theta})\right]\,. \end{aligned}$$

If the complete-data log-likelihood is an exponential family, the maximization in the M-step depends only on a a fixed number of sufficient statistics of $\tilde{P}^{(t)}(\omega)$. These sufficient statistics may be approximated by drawing a sample from $P(\omega|O, \boldsymbol{\theta}^{(t-1)})$ with a Metropolis-Hastings algorithm. This constitutes a Monte-Carlo EM algorithm [29]. It converges if the Monte Carlo error is kept small enough, which can most easily be achieved by increasing the Monte Carlo sample size with every iteration — either with predetermined increments or with data-driven strategies as suggested by [5].

5 Experiments

We validate our method on a real-world dataset from Avito, the largest Russian classifieds site with 70 million unique monthly visitors. This dataset was previously used for a data prediction contest and is publicly available[1]. Users of Avito browse the site and search for products and services in different categories, filtered by location, and sometimes with keywords. Each row of the dataset is a single search action, with columns including a timestamp, indicators of the category, location, and keywords used in the search, as well as anonymous identifiers of the device model, browser version, and browser User-Agent of the client device. In addition, for every search a user identifier is given, obtained from cookies

[1] https://www.kaggle.com/c/avito-context-ad-clicks/data

stored on their machine. The dataset contains a sample of 4.3 million users and spans from April 25th, 2015, to May 20th, 2015. In this period these users made 112 million searches.

For every user we constructed a device fingerprint from the user's device type and model (e.g. Samsung GT-i9500, iPhone, etc.) and browser family (Chrome, Safari, etc.). A total of 9252 distinct fingerprints were found, among 4.3 million unique users. The distribution of fingerprints across users, shown in Fig. 2, is highly skewed: the ten most common fingerprints account for 92% of users.

While these device fingerprints are far from uniquely identifying, we can still use them in combination with other information to discover correspondences between searches. Our goal in this section is to estimate correspondence probabilities for searches by using these partial fingerprints in combination with three models: (1) a model that only considers the number of searches made by each user, (2) the same model, but with one extra parameter left to be estimated from the data with MCMC-EM, and (3) a model that aims to exploit consistency in the sequence of searches made by a user.

All experiments are run on a test set consisting of all the searches made by a sample of 5000 users. These users made a total of 124331 searches with 218 unique fingerprints. Among those fingerprints, 55 were unique to a single device in the test dataset; the most common fingerprint was observed with 1878 different devices. The number of searches made by users follows a heavy-tailed distribution with a median of 5 and a maximum of almost 2000 searches.

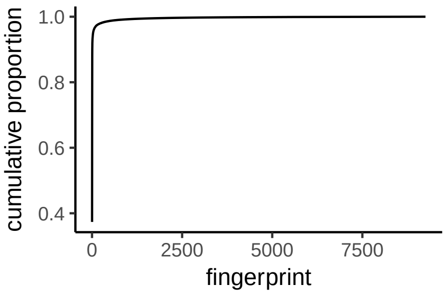

Fig. 2. Distribution of partial fingerprints

5.1 Evaluation metrics

We evaluate the performance of our method with the three models based on the estimated pairwise correspondence probabilities $P(o_i \sim o_j), i \neq j, f_i = f_j$. Our first performance metric is the Brier Score, calculated as

$$BS = \frac{1}{\#\text{pairs}} \sum_{i \neq j,\ f_i = f_j} (\hat{p}_{ij} - y_{ij})^2$$

where $y_{ij} = 1$ if $o_i \sim o_j$ and $y_{ij} = 0$ if $o_i \not\sim o_j$, $\hat{p}_{ij}$ is the estimated correspondence probability for o_i and o_j, and the summation runs over all observation pairs that have the same partial fingerprint. Since there are too many such pairs to enumerate (namely $N(N-1)/2$), we approximate the sum using a sample of 5000 observation pairs. We draw this sample in two different ways: (1) by drawing observations pairs u.a.r. from all possible pairs, and (2) by first drawing one observation u.a.r. and then another with the same fingerprint. Since the number of possible pairs for a fingerprint with n observations is $n(n-1)/2$ rather than n, the first sample will contain many more pairs from the most common fingerprints than the second sample.

The second performance metric is the Logarithmic Loss

$$LL = \frac{-1}{\#\text{pairs}} \sum_{i \neq j,\ f_i = f_j} \left(y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - \hat{p}_{ij}) \right) .$$

The summation is approximated using the same two samples of 5000 pairs. From an information-theoretic perspective, this logarithmic loss is essentially minus the expected cross-entropy between the estimated correspondence probabilities and the true correspondences. A higher log-loss indicates a higher expected surprisal.

5.2 Models and implementation

We now outline the three models and some of the non-trivial calculations involved in their implementation.

Model 1 The first model can be written as

$$\begin{aligned} &N \sim \mathrm{U}(1, S), \qquad \mathbf{f} \sim \mathrm{CRP}(A, d), \\ &|T_i| - 1 \sim \ \mathrm{NegativeBinomial}(r, p), \quad i = 1, ..., N \end{aligned} \tag{9}$$

where S is assumed to be large enough. In practice, S always cancels out of the Metropolis-Hastings acceptance probability. Plugging these distributions into the complete-data likelihood (6) gives

$$\begin{aligned} P(Y|M_1, A, r, p) = \frac{1}{S-1} N! \times \frac{\Gamma(A)\, A^{|F|}}{\Gamma(A+N)} \prod_{f \in F} \Gamma(n_f) \\ \times \prod_{i=1}^{N} \left(\binom{|T_i| + r - 2}{|T_i| - 1} (1-p)^r \, p^{|T_i| - 1} \, |T_i|! \right) . \end{aligned}$$

The parameters A, r and p are estimated from the rows in the data set that are not in the test sample. Using maximum-likelihood, we found the values $\hat{r} = 0.26$ and $\hat{p} = 0.012$. The fingerprint distribution parameters A and d were found with the following procedure. First, we estimated the empirical relationship between the number of users N and the number of unique fingerprints $|F|$ by subsampling N users for a range of N values and each time counting the number of unique fingerprints. Then we fitted the distribution $\mathbf{f} \sim \mathrm{CRP}(A, d)$ by matching the expected number of fingerprints $E(|F|; A, d)$ to the empirical estimates using least squares.

Model 2 The second model is similar to the first model, but instead of a uniform prior on the total number of observed users N a binomial prior is used, i.e. $N \sim \mathrm{B}(S, \rho)$ where $S \gg N$ is fixed at some large value. We estimate ρ from the incomplete-data O using the MCMC-EM algorithm described in Sec. 4.2.

In the M-step we set

$$\begin{aligned}\rho^{(t)} &= \arg\max_{\boldsymbol{\theta}} \mathbf{E}_{\tilde{P}^{(t)}}[\log P(\omega, O|\boldsymbol{\theta})] \\ &\approx \arg\max \sum_{m=1}^{M} \left[\log P(O, \omega^{(m)}|\rho)\right] \\ &= \frac{1}{M} \sum_{m=1}^{M} \frac{N^{(m)}}{S}\end{aligned}$$

where $\omega^{(1)}, ..., \omega^{(M)}$ are MCMC samples drawn in the E-step. The advantage of this model is that prior information on ρ can be used to get better results and faster convergence of the MCMC algorithm.

Model 3 The third model is an extended version of Model 1. This model is designed to exploit coherence in the searches that users make. Recall that the data set includes for every observation an identifier for the location used in the search; users tend to repeatedly use a small number of locations in all their searches. Let z_j^i be the location used in search o_j^i by user i. We assume, for a trajectory T_i, that the associated sequence of searches $\mathbf{z}^i = (z_1^i, z_2^i, ..., z_{|T_i|}^i)$ follows a Dirichlet-Categorical distribution

$$P(\mathbf{z}^i|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(|T_i| + \sum_{k=1}^{K} \alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(n_{ik} + \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)}$$

where $n_{ik} = \sum_{j=1}^{|T_i|} I[z_j^i = k]$. A draw from this distribution can be generated by first drawing a probability vector $\mathbf{p}$ from a Dirichlet distribution with parameter vector $\boldsymbol{\alpha}$ and then drawing $|T_i|$ observations from a categorical distribution with probability vector $\mathbf{p}$. The conditional probability distribution of the location z_j^i used in the j'th search by user i given this user's previous searches equals

$$P(z_j^i = k|z_1^i, ..., z_{j-1}^i) = \frac{\alpha_k + n_{ik}^{j-1}}{\sum_{k=1}^{K} \alpha_k + (j-1)}$$

where n_{ik}^{j-1} is the number of previous searches made with the same location. Since the conditional probability of location k increases with the number of times it has been used before, this model incorporates the idea that users tend to keep using the same locations.

As before, the parameter vector $\boldsymbol{\alpha}$ was estimated a priori from the data excluding the test sample.

Implementation details For all models, data preprocessing was done in R, and core computations were implemented in C++ functions which were called from R using the Rcpp module. Code is available online[2].

For the third model, to get satisfactory acceptance rates an additional type of Metropolis transition was used that extracts or absorbs a group observations with the same LocationID from a user's trajectory in one go.

For all three models a total of 250×10^6 MCMC iterations was used, starting from a randomly initialized assignment $\omega^{(0)}$. For the second model, these iterations were spread out over 250 MCMC-EM iterations. Convergence was confirmed by manual inspection. The performance metrics were then computed with another 50×10^6 samples. Running on a machine with a 2.0 GHz Intel Xeon CPU and plenty of memory (the algorithm needs about 5GB with this dataset), the MCMC algorithm took about ten seconds per 10^6 samples.

5.3 Results

Table 1 shows the performance of the three models. As a benchmark this table also shows the scores obtained by naively assuming that observations with matching fingerprints are always made by the same user.

Table 1. Calibration of the correspondence probabilities estimated on the test dataset using the three models and a naive benchmark model which assumes observations with matching fingerprints correspond to one device with probability one. Lower scores mean better calibration. The metrics on the first and third row are computed on a sample of observations pairs randomly drawn from the set of all possible pairs with matching fingerprints. The metrics on the second and fourth row are computed on a stratified sample in which the number of pairs with fingerprint f is proportional to n_f.

	naive	model1	model2	model3
Brier Score	0.925	0.0390	0.0366	0.0224
Brier Score (weighted)	0.945	0.0313	0.0309	0.0150
Log-loss	12.8	0.249	0.222	0.117
Log-loss (weighted)	13.1	0.216	0.222	0.0919

All three models give calibrated correspondence estimates. This can also be seen in the calibration plots in Fig. 3. Results for model 2, in which an extra model parameter was estimated from the incomplete-data, are comparable with

[2] https://github.com/michaelciere/partial-fingerprints

the results from model 1. The correspondence probability estimates obtained by model 3, which attempts to capture coherence in the locations by which users filter their searches, are markedly better than those for the first two models. This improvement is larger for the weighted scores, which can be explained by the fact that the most common fingerprints, which have a smaller influence on the weighted scores, are shared by so many users that estimating correspondences remains difficult even when locations are taken into account. In addition, for these common fingerprints MCMC sampling is difficult and many iterations are needed for the correspondence probability estimates to converge.

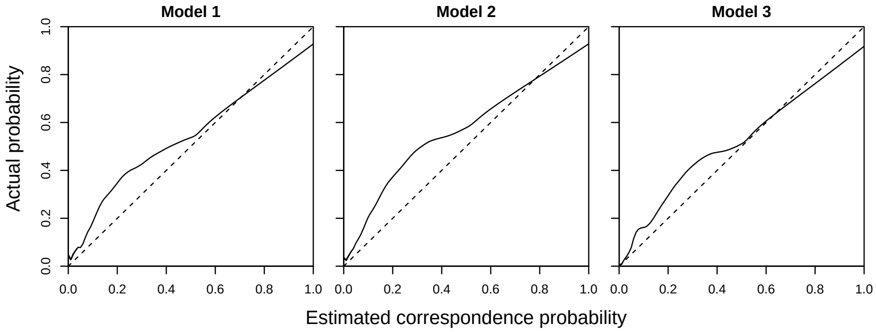

Fig. 3. LOWESS-smoothed calibration curves of estimated correspondence probabilities.

6 Conclusions

This paper has introduced a general-purpose probabilistic method for identifying devices from non-unique device fingerprints. Our experiments have shown that this method produces calibrated correspondence probability estimates. Especially promising are the results obtained by combining partial device fingerprints with user modeling. Many challenges remain. For instance, we have assumed in this work that the device fingerprints are stable, meaning they are measured without noise. When fingerprints are noisy, this noise has to be modeled as well, which introduces additional complications. Furthermore, online deployment of this method would require one to model the arrival and departure of devices. Resolving these issues opens the door to new uses of device fingerprinting in fraud detection, content personalization, and automated authentication.

References

[1] Antoniak, C.E.: Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. The Annals of Statistics 2(6), 1152–1174 (Nov 1974), http://projecteuclid.org/euclid.aos/1176342871

[2] Bar-Shalom, Y., Fortmann, T.E.: Tracking and Data Association. Academic Press (1988)

[3] Blackman, S.S.: Multiple-target Tracking with Radar Applications. Artech House (Jan 1986)

[4] Brik, V., Banerjee, S., Gruteser, M., Oh, S.: Wireless Device Identification with Radiometric Signatures. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking. pp. 116–127. MobiCom '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1409944.1409959

[5] Caffo, B.S., Jank, W., Jones, G.L.: Ascent-based Monte Carlo expectation maximization. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2), 235–251 (Apr 2005), http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2005.00499.x/abstract

[6] Cox, I.J.: A review of statistical data association techniques for motion correspondence. International Journal of Computer Vision 10(1), 53–66 (1993), http://link.springer.com/article/10.1007/BF01440847

[7] Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. Machine Learning 50(1-2), 45–71 (Jan 2003), http://link.springer.com/article/10.1023/A:1020245811187

[8] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977), http://www.jstor.org/stable/2984875

[9] Dorazio, R.M., Royle, J.A., Sderstrm, B., Glimskr, A.: Estimating species richness and accumulation by modeling species occurrence and detectability. Ecology 87(4), 842–854 (Apr 2006)

[10] Eckersley, P.: How Unique is Your Web Browser? In: Proceedings of the 10th International Conference on Privacy Enhancing Technologies. pp. 1–18. PETS'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1881151.1881152

[11] Ewens, W.J.: The sampling theory of selectively neutral alleles. Theoretical Population Biology 3(1), 87–112 (Mar 1972), http://www.sciencedirect.com/science/article/pii/0040580972900354

[12] Gilks, W.R.: Markov Chain Monte Carlo. In: Encyclopedia of Biostatistics. John Wiley & Sons, Ltd (2005), http://onlinelibrary.wiley.com/doi/10.1002/0470011815.b2a14021/abstract

[13] Huang, T., Russell, S.: Object Identification in a Bayesian Context. In: Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2. pp. 1276–1282. IJCAI'97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997), http://dl.acm.org/citation.cfm?id=1622270.1622340

[14] Kohno, T., Broido, A., Claffy, K.C.: Remote physical device fingerprinting. IEEE Transactions on Dependable and Secure Computing 2(2), 93–108 (Apr 2005)

[15] Luo, J., Pattipati, K.R., Willett, P.K., Hasegawa, F.: Near-optimal multiuser detection in synchronous CDMA using probabilistic data association. IEEE Communications Letters 5(9), 361–363 (Sep 2001)

[16] Marinakis, D., Dudek, G.: Occam's Razor Applied to Network Topology Inference. IEEE Transactions on Robotics 24(2), 293–306 (Apr 2008)

[17] Marinakis, D., Dudek, G., Fleet, D.J.: Learning Sensor Network Topology through Monte Carlo Expectation Maximization. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. pp. 4581–4587 (Apr 2005)

[18] Marinakis, D., Dudek, G.: Topological Mapping Through Distributed, Passive Sensors. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. pp. 2147–2152. IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007), http://dl.acm.org/citation.cfm?id=1625275.1625622

[19] Mowery, K., Bogenreif, D., Yilek, S., Shacham, H.: Fingerprinting Information in JavaScript Implementations. ResearchGate (May 2012)

[20] Neal, R.M., Hinton, G.E.: Learning in Graphical Models. pp. 355–368. MIT Press, Cambridge, MA, USA (1999), http://dl.acm.org/citation.cfm?id=308574.308679

[21] Neumann, C., Heen, O., Onno, S.: An Empirical Study of Passive 802.11 Device Fingerprinting. In: 2012 32nd International Conference on Distributed Computing Systems Workshops. pp. 593–602 (Jun 2012)

[22] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., Vigna, G.: Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. In: 2013 IEEE Symposium on Security and Privacy (SP). pp. 541–555 (May 2013)

[23] Oh, S., Russell, S., Sastry, S.: Markov Chain Monte Carlo Data Association for Multi-Target Tracking. IEEE Transactions on Automatic Control 54(3), 481–497 (Mar 2009)

[24] Oh, S., Russell, S., Sastry, S.: Markov chain Monte Carlo data association for general multiple-target tracking problems. In: 43rd IEEE Conference on Decision and Control, 2004. CDC. vol. 1, pp. 735–742 Vol.1 (Dec 2004)

[25] Pasula, H., Russell, S.J., Ostland, M., Ritov, Y.: Tracking Many Objects with Many Sensors. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. pp. 1160–1171. IJCAI '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999), http://dl.acm.org/citation.cfm?id=646307.687451

[26] Pitman, J.: Combinatorial Stochastic Processes, Lecture Notes in Mathematics, vol. 1875. Springer-Verlag, Berlin/Heidelberg (2006), http://link.springer.com/10.1007/b11601500

[27] Sittler, R.W.: An Optimal Data Association Problem in Surveillance Theory. IEEE Transactions on Military Electronics 8(2), 125–139 (Apr 1964)

[28] Spooren, J., Preuveneers, D., Joosen, W.: Mobile Device Fingerprinting Considered Harmful for Risk-based Authentication. In: Proceedings of the Eighth European Workshop on System Security. pp. 6:1–6:6. EuroSec '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2751323.2751329

[29] Wei, G.C.G., Tanner, M.A.: A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. Journal of the American Statistical Association 85(411), 699–704 (Sep 1990), http://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10474930