Probabilistic Visitor Stitching on Cross-Device Web Logs

Sungchul Kim
Adobe Research
San Jose, CA 95110
sukim@adobe.com

Nikhil Kini
UC Santa Cruz
Santa Cruz, CA 95064
nkini@ucsc.edu

Jay Pujara
UC Santa Cruz
Santa Cruz, CA 95064
jay@cs.umd.edu

Eunyee Koh
Adobe Research
San Jose, CA 95110
eunyee@adobe.com

Lise Getoor
UC Santa Cruz
Santa Cruz, CA 95064
getoor@soe.ucsc.edu

ABSTRACT

Personalization – the customization of experiences, interfaces, and content to individual users – has catalyzed user growth and engagement for many web services. A critical prerequisite to personalization is establishing user identity. However the variety of devices, including mobile phones, appliances, and smart watches, from which users access web services from both anonymous and logged-in sessions poses a significant obstacle to user identification. The resulting entity resolution task of establishing user identity across devices and sessions is commonly referred to as "visitor stitching." We introduce a general, probabilistic approach to visitor stitching using features and attributes commonly contained in web logs. Using web logs from two real-world corporate websites, we motivate the need for probabilistic models by quantifying the difficulties posed by noise, ambiguity, and missing information in deployment. Next, we introduce our approach using probabilistic soft logic (PSL), a statistical relational learning framework capable of capturing similarities across many sessions and enforcing transitivity. We present a detailed description of model features and design choices relevant to the visitor stitching problem. Finally, we evaluate our PSL model on binary classification performance for two real-world visitor stitching datasets. Our model demonstrates significantly better performance than several state-of-the-art classifiers, and we show how this advantage results from collective reasoning across sessions.

Keywords

Visitor stitching; Cross-device users; Personalization

1. INTRODUCTION

Ubiquitous computing has transformed the landscape of how society interacts with web services. A single user will often access web services from a wide range of devices, including desktop and laptop computers at both home and work, tablets, mobile devices, vehicles, and entertainment systems. Across these varied devices, users expect services to remember their preferences and provide a seamless user experience and interface. However, users frequently access these services from a mixture of authenticated and anonymous sessions, making it difficult to identify the user and provide a tailored experience. The problem of consolidating multiple visits across different devices and sessions into a single user identity is known as visitor stitching.

Traditionally, web services have relied on cookies to identify users. However, in two real-world datasets we examine, over half of the users have multiple cookie identifiers. This problem has been documented in a number of research studies. Dasgupta et al. [8] demonstrate that users often possess more than one cookie identifier and Coey et al. [6] showed that in an online experiment with treatment and control groups, cookie-level assignment resulted in imperfect design, and has the potential to under-estimate the true treatment effects. In fact, users may not only possess multiple cookie identifiers, but they may also have identifiers across multiple devices, browsers, or even share them between different users. For IT companies providing large-scale web services, stitching together web logs belonging to unique users across several sources is a crucial barrier to accurately estimating behaviors and statistics at the user level.

Typical approaches to solving the visitor stitching task rely on proprietary information specific to a particular domain, such as search behavior, purchase history, or topical and content information [5, 9, 8]. A related problem, identifying the same user across social networks [15, 30], has also been solved using proprietary information, features specific to social networks, and domain-specific problem formulations, such as bipartite matching. The success of these approaches demonstrates the promise of visitor stitching. However, the reliance on proprietary features and problem settings makes it difficult to generalize these contributions across a broader set of applications. In this paper, we enumerate features universally available in web logs, and perform an analysis of the discriminative power of these features using real-world data from two different companies.

One conclusion of our analysis is that web log features inherently vary widely in discriminative power. We propose a probabilistic approach that is capable of learning the reliability of web log features and combining these features to improve discriminative power. Our solution utilizes probabilistic soft logic (PSL) [1], a popular statistical relational learning framework, to construct a general-purpose model

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052711

for the visitor stitching problem that is effective across domains. PSL enables scalable development of probabilistic models for reasoning about relational, structured, and uncertain data. PSL models use logical rules to represent the potential functions of a powerful class of probabilistic graphical models known as Hinge-loss Markov random fields (HL-MRFs). For visitor stitching, PSL allows us to overcome any lack in discriminative power from the information in noisy web logs by constructing a flexible probabilistic structure over features (e.g., IP addresses, device types, etc.), ultimately enabling joint inference of same-visitor web logs in a manner that respects that structure.

We first introduce the general visitor stitching setting by describing our real-world web logs and provide an entropy-based analysis to demonstrate the lack of strong discriminative power in common features of the naturally noisy and complex logs. Next, we address these issues of uncertainty and propose several PSL models including base models that rely on singular features, as well as a final, more complex model that leverages the generally available features to collectively determine sets of web logs belonging to the same user. Finally, we apply our PSL models on two real-world datasets and demonstrate their effectiveness on the binary classification task of identifying web logs belonging to the same user. On this task, compared to the state-of-the art classifiers, our PSL models achieve statistically better F-scores reaching 0.971 and 0.837 for two datasets, respectively. We also show how PSL allows collective reasoning to propagate information about user identity; we show that, in the semi-supervised setting, where we are given partial information about user matches, we can use this to infer remaining user pairs.

The remaining content of the paper is organized as follows: Section 2 provides an introduction to related attempts at addressing the visitor stitching problem. Section 3 describes the real-world web logs data and provides an analysis of common features based on normalized entropy. Section 4 is a short introduction to the concept of PSL. Section 5 defines the visitor stitching problem statement and describes in more detail the PSL models used in the solution. Section 6 provides the experimental details and results of these models. Finally, we discuss our results and additional considerations on the visitor stitching problem.

2. RELATED WORK

In this section, we highlight some recent advances in visitor stitching, and work in using information about a user's geographic location.

2.1 Visitor Stitching with Web Logs

There are several prior approaches that have analyzed web cookie or session logs for the visitor stitching task. Eckersley et al. [9] proposed a fingerprint method that uses a number of web browser features including a user-agent string, HTTP header, plugin information, screen resolution, and more. They provide precise statistics and analysis results of web browsers, and it paves the way using it for user identification. However, the independence assumption of the browser features needs to be carefully handled [8]. Dasgupta et al. [8] introduced an alternate clustering method to address the cookie churn problem. They introduce a 'cannot-link' constraint that assumes the lifetime of cookies cannot overlap. However, there can be counter examples to this assumption, especially in multi-device and multi-browser environments where users may simultaneously access services. More recently, Saha et al. [24] attempted to solve this task by viewing the problem as a binary classification task and utilizing classification methods based on several visit features, as well as a 'User ID' feature provided by Google Universal Analytics to connect multiple devices and sessions [11].

Apart from the standard visitor stitching task, there has also been work done on resolving users on shared devices. Based on search-logs, Montanez et al. [19] attempt to predict the transition between devices of each user by analyzing searches across devices. Even more recently, White et al. [28] suggested methods to predict the presence of multiple searchers associated with a machine identifier, and proposed a clustering method via users' historical data that accurately assigns new users to the correct existing user. Finally, Casado et al. [5] proposed measurement techniques, and implemented a method to quantify the usage of NATs, dynamic addressing, and proxies on IP addresses to give insights of whether or not IP address is a useful identifier.

As previously mentioned, most prior work [17, 18, 29] typically uses proprietary information specific to a particular domain such as search behavior, purchase history, and topical and content information. Though this information can be helpful to enhance performance, it is unlikely to be universal across domains.

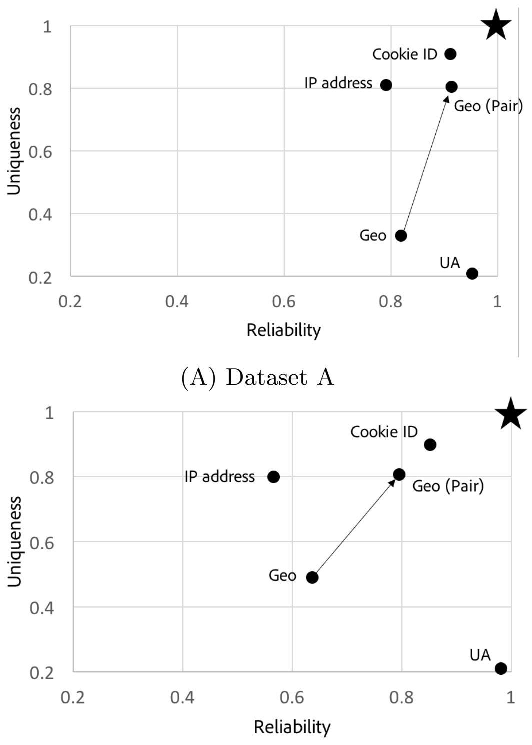
2.2 Utilizing User Geo-location

Identifying a user's geographic location is one of the most important and commonly used techniques in applications focused on enhancing user experience [3, 7]. Accordingly, in the field of information retrieval, Backstrom et al. [2] focus on local aspects of search queries and introduce a generative model for describing local properties of queries including observational studies on which queries highly correlated with user location. They mostly evaluate the quality or confidence of the geo-location for a new user, which can be used in real-world applications such as targeted advertising or visitor stitching. More recently, based on the time-stamped location data, Riederer et al. [23] suggested a method to identify the same user across domains. However, their work focuses on the mobile users only with geo location and time information, while our approach attempts to link cross devices web logs.

From a different view point, Liben-Nowell et al. [13] provide observations extracted from the analysis of geographic and social proximity. For example, their observational study shows that the likelihood of friendship is inversely proportional to distance between persons; after exceeding a certain distance the relationship belongs to a baseline probability that is entirely independent of geographic location. Similarly, Backstrom et al. [3] propose an algorithm that accurately locates users, and address the interplay between geographic distances and social relationships. Although the social information they utilize is available only via social network data, their work could be helpful for visitor stitching since social information, as they discuss, can more accurately detect location than IP addresses and is positively correlated with individual website usage patterns.

3. DATA ANALYSIS

In this section, we present analysis results on two real-world datasets. The data describes clickstream logs over



(A) Dataset A

(B) Dataset B

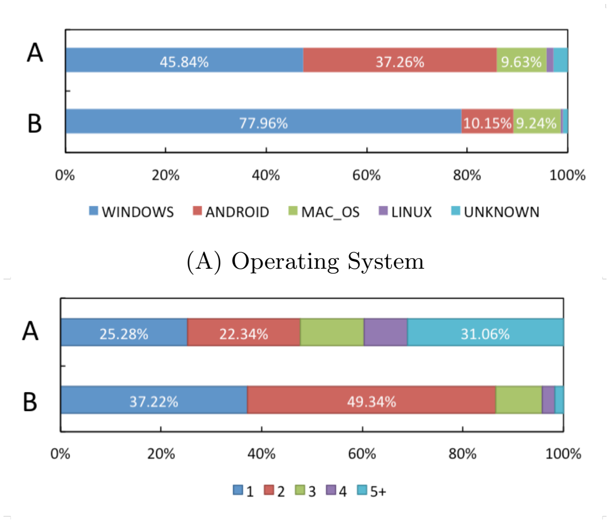
Figure 1: Plots of features according to entropy-based measures; the star mark represents the user identifier.

a two-week period in January 2015 from Company A and Company B, which have 7.3 million and 2.2 million user accounts, respectively. We remove user accounts that have more than 20 cookie identifiers, which are likely attributed to bot-like activities. Additionally, we filter out user accounts with only one cookie identifier, to focus on users who access services multiple times and enable an adequate testing environment. Post-filtering, the number of multiple-cookie owning users is 31K for dataset A, and 270K for dataset B[1].

We also focus on the feature of geographic locations of users, describing how we make use of geo-coordinate sets to simultaneously simplify models and provide this discriminative power for the visitor stitching problem in the following sub-sections.

3.1 Entropy-based Feature Analysis for Visitor Stitching

We perform a feature analysis of available information in the web logs in order to quantify the discriminative power of features available. We define two metrics to assess these features. The first is **uniqueness**, which is based on the entropy over unique *users* who share a feature value. The second metric is **reliability**, based on the entropy of unique *feature values* observed for a single user. We assume that high-quality features should consistently identify only one user (uniqueness) and have consistent values across a user's visits (reliability).



(A) Operating System

(B) Cardinality of IP address per user

Figure 2: Stacked bar graph of user distribution in features

The entropy-based evaluation measures are computed based on the normalized entropy as follows:

$$1 - H(x) = 1 - \sum_{y_i \in Y} -p_{y_i}(x) \log p_{y_i}(x) \qquad (1)$$

where $p_{y_i}(x) = |x|_{y_i} / (\sum_{x_j \in X_{y_i}} |x_j|_{y_i})$.

To measure uniqueness, for each feature value (y_i) in the set of all feature values (Y) we generate a set of associated users (X_{y_i}) and compute the entropy. We then take the average of the entropy values across features to determine the uniqueness entropy. Computing reliability entropy inverts this process. For each user (y_i) in the set of all users (Y), we generate a set of that user's feature values (X_{y_i}) and compute a user-specific feature entropy, and then average across users to get a reliability entropy. A uniqueness entropy of 1 indicates that each feature value is associated with a single user, while a reliability entropy of 1 indicates that each user is associated with a single feature value. Lower values indicate that a feature is associated with many users (for uniqueness) or that a user has many distinct feature values (for reliability).

Based on these two entropy-based measures, we can obtain insights into which features possess stronger predictive signal for stitching web logs (Figure 1). We consider four features: cookie identifier, user-agent strings, IP addresses, and geographic coordinates. These features were selected since they are consistently available across all cookie logs, whereas other features are specific to the corresponding website or service that is monitoring activity. In this figure, the star located at (1,1) represents the user identifier, which is the label in our dataset. Intuitively, cookie identifier is very close to user identifier, but not a perfect match. Since a cookie is unique to a browser instance (until it gets churned), it does not possess a signal to stitch multiple web logs from the same user.

3.2 User-Agent Strings & IP Addresses

The user-agent string contains information about the browser and the operating system of the user, including the ver-

[1] The ratio of single-cookie users in company A is much larger than company B

sion, the plugin, and much more. Websites can utilize this information to provide content tailored for users' specific browsers. According to the analysis, dataset A contains more variations in devices (Figure 2 (A)) and IP Addresses per visitor (Figure 2 (B)) relative to dataset B. In addition, it shows that up to 40% of users are using mobile devices. In terms of the entropy measures, the user-agent string shows high reliability, because individual users generally use a small set of devices and browsers. However, it shows the low uniqueness, since many users will have identical user-agent strings (i.e., use the same browser and operating system). IP addresses show the opposite pattern. IP addresses have low reliability, since users access services from multiple network (in our dataset over 60% of visitors had multiple IP addresses), but high uniqueness, since a single network (such as a home) will have very few users. We exploit this complementary relationship in our model by pairing the user-agent string and IP address together when performing visitor stitching.

3.3 Geographic Locations

Among the many possible geo-location features including country, state, city, zip code, latitude, and longitude, we consider the most precise feature – latitude and longitude, and refer to it as the geo-coordinate. Users naturally have more than one geo-coordinate (if (s)he has mobile devices, it varies more), and it is reasonable to use a set of geo-coordinates to represent a user's position as a feature.

In Figure 1, the geo-coordinate contains less uniqueness and reliability information than the other features that we consider. However, by leveraging people's behavior of frequent visits to their workplace and home, we simply collapse the top-two most visited locations into a pair of coordinates. We then re-computed the entropy values using this frequent geo-coordinate pair, and as seen in the figure, it is much closer to the user identifier. It shows that frequently visited geo-coordinates contain very unique and reliable information about users. In addition, in contrast to other features, geo-coordinates can be used to compute the distance between cookie pairs. For example, if two visits are detected in very close proximity, the separate IP addresses would imply separate users; however, using geo-location, we can more confidently infer if the visits come from the same underlying user.

3.4 Geo-Core selection

User geo-coordinate lists can be viewed as noisy sets with clusters surrounding frequently visited explainable locations such as a workplace or home. In order to properly stitch visits together, we are interested in uncovering the core set of geo-coordinates that indicate where a user connected to a service. Unfortunately, we cannot use typical clustering methods that find coarse-grained clusters from a large number of data points, since we would need to apply this process separately for each user. In web logs, each user has their own regions from where they frequently access the website, and their geo-coordinates are grouped in a few dense and consistent regions with some noise. The primary goal of core geo-coordinate selection is to find geo-coordinates of those regions while minimizing the noise. To do this, we bucket visits based on the time segments and merge nearby coordinates to reduce noise.

Formally, given a list of geo-coordinates X in which each geo-coordinate is obtained from a user's web logs, we do the following:

1. Divide X into equal-sized, temporally partitioned buckets, where the l-th bucket is $X_l = \{x_{1_l}, x_{2_l} \ldots x_{n_l}\}$.
2. For each bucket l:
   (a) Compute the density of each geo-coordinate x_{i_l} as $f(x_{i_l}) = \sum_{x \in X_l} I(dist(x, x_{i_l}) < \theta)/|X_l|$.
   (b) Merge geo-coordinates within a distance of θ. Specifically, in a set of merged geo-coordinates, the one with the highest density is retained and others discarded.
3. Cluster the resultant geo-coordinates from all buckets into k clusters and select their centers.

θ was determined based on the average distance between each geo-coordinate with its 1-nearest geo-coordinate for the same user[2]. In step 2b, if two geo-coordinates have the same density and are within θ of each other, they are merged and any one of them will be used to represent the merged set. We use conventional k-means clustering with $k = 3$ to indicate home, work, and one additional primary region[3].

3.5 Measuring Geolocation Distance

To utilize the user geo-coordinates for the visitor stitching task, we need to define a similarity measure between users which allows each user to have a variety of associated geo-coordinates. Accordingly, we need to compute accurate and reliable distance between geo-coordinate sets regardless of whether it has one or many geo-coordinates. In this work, we use a variation of the minimum bipartite matching [27] between geo-coordinate sets. Formally, given a bipartite graph $G = (U, V, E)$, where its vertices are divided into two disjoint sets U and V and each edge (u_i, v_j) indicates a connection between them, the minimum bipartite matching can be formalized as follows:

$$\begin{aligned}
&\min_{E} \sum_{(u_i, v_j) \in E} w_{ij} x_{ij} \\
\text{s.t.} &\sum_{i=1}^{N} x_{ij} = 1 \quad \forall j = 1, \cdots, N \\
&x_{ij} \in \{0, 1\}
\end{aligned}$$

where N is the smaller cardinality between two geo-coordinate sets $\min(|U|, |V|)$, w_{ij} is distance between geo-coordinates, and x_{ij} is 1 if (u_i, v_j) is an edge of the minimum weighted bipartite matching.

4. BACKGROUND

Our approach to visitor stitching uses the probabilistic soft logic framework (PSL) to construct a graphical model

[2] θ was determined as 1 mile (1.6 km) based on the average distance between each geo-coordinate and its 1-nearest neighbor (1.3157 km and 1.5952 km in dataset A and B, respectively)

[3] Although k is 3, the resultant number of clusters can be less than 3 when certain clusters do not have instances. In addition, when k is larger than 3, the performance generally decreases.

using web log features. PSL is an increasingly popular modeling tool that has been successfully applied to many domains including collective inference [14, 20], ontology alignment [21], personalized medicine [10], and recommender systems [12].

PSL models are specified through rules expressed in first order logic syntax. However, the atoms in PSL rules take values in the $[0, 1]$ interval. For example, consider the PSL rule,

$$\begin{aligned}&\textsc{VisitorIP}(\mathtt{V}_1, \mathtt{I}_1) \wedge \textsc{VisitorIP}(\mathtt{V}_2, \mathtt{I}_2) \wedge \textsc{SimIP}(\mathtt{I}_1, \mathtt{I}_2)\\ &\implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)\end{aligned} \tag{2}$$

This rule specifies that users with similar IP addresses are likely to be the same user. The VisitorIP predicate associates a user with an IP address, while the SimIP predicate captures the similarity of two IP addresses, and can have values between 0 and 1. The inferred variable, $\textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$, similarly takes values in $[0, 1]$, expressing the confidence of the inference.

Each PSL rule is associated with a weight, and has a *truth value* which can be derived using the Lukasiewicz t-norm and co-norm:

$$\begin{aligned} a \ \wedge \ b &= \max\{a + b - 1, 0\}\\ a \ \vee \ b &= \min\{a + b, 1\}\\ \neg\, a &= 1 - a \end{aligned} \tag{3}$$

where $a, b \in [0, 1]$ are soft truth values or similarity scores, and the Lukasiewicz norm and co-norm are used as an alternative to Boolean operators.

PSL restricts rules to be a disjunction of literals, which allows each rule to be translated into a hinge-loss potential of the form: $\phi_j(X, Y) = \max(0, l_j(X, Y))$, where l_j is a linear function of random variables Y and evidence X. Each ground rule in the PSL model corresponds to a distinct weighted potential, and together these potentials define a joint probability distribution over the variables:

$$P(Y|X) = \exp\left[-\sum_{j=1}^{m} w_j \phi_j(X, Y)\right] \tag{4}$$

The resulting models, known as hinge-loss Markov random fields (HL-MRFs), admit a convex inference objective, allowing for efficient MAP inference.

Using PSL models, we can take advantage of its extensibility of PSL. Our models are specified using a set of PSL rules, which describes the relationships among cookies (or users). Thus, if additional domain-specific information becomes available, we can define new rules to include it. For example, if browsing history is available, we could exploit the following rule:

$$\begin{aligned}&\textsc{BrowsingLog}(\mathtt{V}_1, \mathtt{L}_1) \wedge \textsc{BrowsingLog}(\mathtt{V}_2, \mathtt{L}_2)\\ &\wedge \textsc{SimLog}(\mathtt{L}_1, \mathtt{L}_2) \implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)\end{aligned}$$

where BrowsingLog() is a list of pages that each user has visited and SimLog() is a user-defined function that returns the similarity between two sets of browsing histories.

5. PROPOSED PROBABILISTIC APPROACH

In this section, we formally define the visitor stitching task and enumerate the models used in our probabilistic approach for solving the problem.

5.1 Problem Statement

We define a visitor based on a set of web log entries corresponding to a unique identifier, such as as a cookie or device identifier. The visitor stitching problem can be formulated as a binary classification task, where the goal is to predict whether two visitors in a web log are the same user. For each pair of visitors, $\mathtt{V}_1, \mathtt{V}_2$, a classifier will predict 1 if these visitors are the same user and 0 if they are distinct users.

In the general visitor stitching setting, the most commonly available pieces of information recorded in web logs include the visitor's IP address, geographic location, and user-agent string. URL visit or product purchase information is often available as well. However, we found that such information does not have much predictive power for stitching visits with our current datasets, because they are collected from a single company's website and contain visit information for only that URL. URL visits or product purchase information may be more useful if datasets contain information across different sites.

5.2 PSL Models

We begin with three base models which utilize feature similarities: 1) an IP address model (IP), 2) a user-agent model (UA) and 3) a geo-coordinate model (Geo). We then describe our full, collective PSL models that utilize these features in more advanced ways to make use of the relational structure of the data.

Model Prior and Rule Weights: For all models, we introduce a negative prior of the form

$$\neg\textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$$

This captures the prior probability that two visitors are unlikely to be the same user in the absence of other evidence. The weight of this prior, and all rules in the sequel, are learned using training data. Additionally, we use training data to learn an optimal threshold for the soft-truth value of same-user predictions, $\textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$, allowing a binary classification.

IP: Our first model is a baseline model which uses a single signal, IP address similarity between two visitors (V_1 and V_2), to infer whether these visitors are the same user ($\textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$).

$$\begin{aligned}&\textsc{IP}(\mathtt{V}_1, \mathtt{I}_1) \wedge \textsc{IP}(\mathtt{V}_2, \mathtt{I}_2) \wedge \textsc{SimIP}(\mathtt{I}_1, \mathtt{I}_2)\\ &\implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)\end{aligned}$$

where the atom $\textsc{SimIP}(\mathtt{I}_1, \mathtt{I}_2)$ is computed using the Jaccard coefficient between sets of IP addresses associated with the respective visitors.

UA: Our second baseline model also uses a single feature, user-agent strings, to make a prediction.

$$\begin{aligned}&\textsc{UA}(\mathtt{V}_1, \mathtt{U}_1) \wedge \textsc{UA}(\mathtt{V}_2, \mathtt{U}_2) \wedge \textsc{SimUA}(\mathtt{U}_1, \mathtt{U}_2)\\ &\implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)\end{aligned}$$

$\textsc{SimUA}(\mathtt{U}_1, \mathtt{U}_2)$ is computed by first using a hash function to convert the user-agent string into an ID-like string based on operating system and browser info, and also uses the Jaccard coefficient between sets of user-agent strings appearing in different cookie logs.

Geo: Our third baseline is a visitor geo-location (Geo) model based on the following rule, which employs the geographical distance of a two visitors.

$$\textsc{Loc}(\mathtt{V}_1, \mathtt{L}_1) \wedge \textsc{Loc}(\mathtt{V}_2, \mathtt{L}_2) \wedge \textsc{Close}(\mathtt{L}_1, \mathtt{L}_2) \\ \implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$$

where $\textsc{Loc}(\mathtt{V}_i, \mathtt{L}_i)$ is a predicate that indicates that the visitor $\mathtt{V}_i$ is located at $\mathtt{L}_i$ and $\textsc{Close}(\mathtt{L}_1, \mathtt{L}_2)$ is computed as described in Section 3.5, which handles both single-coordinates (e.g., center coordinate, or the most frequently occurring coordinate) or coordinate sets (e.g., all coordinates, or core-coordinates). This raw geographical distance between visits is converted in a range of [0,1] using $f(d) = e^{-kd}$.

IP·UA: More sophisticated rules can combine signals from multiple features. In Section 3, we discussed that the user-agent string has high reliability but suffers from low uniqueness since users choose from a limited set of browsers. Conversely, the IP address has low reliability since users visit from multiple networks, but high uniqueness. Combining these two features overcomes their individual weaknesses.

$$\text{IP}(\mathtt{V}_1, \mathtt{I}_1) \wedge \text{IP}(\mathtt{V}_2, \mathtt{I}_2) \\ \wedge \text{UA}(\mathtt{V}_1, \mathtt{U}_1) \wedge \text{UA}(\mathtt{V}_2, \mathtt{U}_2) \wedge \textsc{SimIPUA}(\mathtt{I}_1, \mathtt{I}_2, \mathtt{U}_1, \mathtt{U}_2) \\ \implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$$

Combining $\textsc{SimIP}(\mathtt{I}_1, \mathtt{I}_2)$ and $\textsc{SimUA}(\mathtt{U}_1, \mathtt{U}_2)$ into one rule enables us to force the rule to ground only under circumstances when the joint similarity is high rather than individually. For visitor identification, we need a strict 'and' operation on IP and UA rules. Thus, we define $\textsc{SimIPUA}()$ on $\mathtt{I}_1$, $\mathtt{I}_2$, $\mathtt{U}_1$, $\mathtt{U}_2$ as $\textsc{SimIP}(\mathtt{I}_1, \mathtt{I}_2) \times \textsc{SimUA}(\mathtt{U}_1, \mathtt{U}_2)$, which is valid only when both the terms are larger than 0.

Geo·UA: Similar to the previous model, Geo·UA model considers user geo-location as the primary factor, and additionally uses the user-agent string as a secondary factor to support the geo-location using a combined similarity function.

$$\textsc{Loc}(\mathtt{V}_1, \mathtt{L}_1) \wedge (\mathtt{V}_2, \mathtt{L}_2) \wedge \text{UA}(\mathtt{V}_1, \mathtt{U}_1) \\ \wedge \text{UA}(\mathtt{V}_2, \mathtt{U}_2) \wedge \textsc{CloseLocUA}(\mathtt{L}_1, \mathtt{L}_2, \mathtt{U}_1, \mathtt{U}_2) \\ \implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2)$$

Like the Geo Model, the similarity between geo-coordinates is defined according to input coordinate type. However, we define $\textsc{CloseLocUA}()$ on $\mathtt{L}_1$, $\mathtt{L}_2$, $\mathtt{U}_1$, $\mathtt{U}_2$ as $\textsc{Close}(\mathtt{L}_1, \mathtt{L}_2) \times \textsc{SimUA}(\mathtt{U}_1, \mathtt{U}_2)$, valid only when both the terms are larger than 0, like we did for $\textsc{SimIPUA}()$ in the IP·UA Model.

Transitivity: In addition to pair-wise relationships between cookies mentioned in previous sections, PSL allows collective inference by incorporating connectivity in relational data. One example is the transitivity rule, written as follows:

$$\textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_2) \wedge \textsc{SameUser}(\mathtt{V}_2, \mathtt{V}_3) \\ \implies \textsc{SameUser}(\mathtt{V}_1, \mathtt{V}_3)$$

By using the transitivity rule, PSL can infer that two visitors $\mathtt{V}_1$and $\mathtt{V}_3$ are the same user, even when the pair-wise similarity between $\mathtt{V}_1$and $\mathtt{V}_3$ is low, via strong connectivity between $(\mathtt{V}_1, \mathtt{V}_2)$ and $(\mathtt{V}_2, \mathtt{V}_3)$.

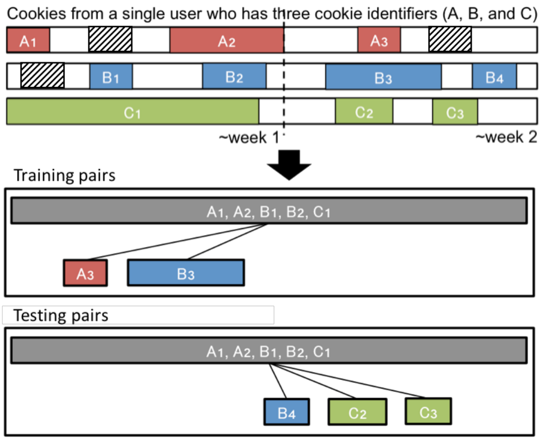


Figure 3: Web log pair extraction for a user with separate cookie identifiers A, B, and C

Additional Models: All remaining models combine features as described in Model IP·UA and Geo·UA. The considerations here extend from the feature analysis in Section 3. Specifically, though geo-coordinate models are likely to show more discriminative power on their own than the other feature-based models, combining coordinates with IP-addresses and user-agent strings in various ways can help capture cookie pairs belonging to the same user that would not otherwise be captured.

6. EXPERIMENTS

In this section, we first provide the details of the experimental setting including data collection, evaluation measures, and pre-processing. Then, we discuss evaluation results for all of the models and demonstrate the effectiveness of modeling the visitor identification problem with PSL.

6.1 Evaluation Results

Data preparation: In our dataset described in Section 3, web logs contain a typical cookie identifier, and also login information if they were recorded after user login. Every visit by a user results in multiple entries in the web logs, so log entries are aggregated to generate visits. For week 1, we aggregate logs based on a user login information. For week 2, logs are aggregated based on their cookie identifier. Then, pairs are generated by pairing a visit from the user log in week 1 and a visit from the cookie log from week 2. The training pairs are random samples from the generated pairs, and testing pairs are those not selected for the training set (Figure 3). The training pairs represent the cases of future web logs with login information. The testing pairs represent web logs without login information, and thus need to be predicted. Overall, there are 211k and 131k pairs are generated in Data A and B, respectively. Since we divide the pairs into training and testing for each user, about half the pairs are used for training and the other half is used for testing. For evaluation, we conduct 10-fold cross validation and report results in terms of the F-score (F1), precision (Pre.), and recall (Rec.).

Models: We evaluate our PSL models and compare against a variety of off-the-shelf classifiers including Naive Bayes (NB), Logistic Model Tree (LM) [25], MultiBoosting (MB),

Dataset	Models	Pre.	Rec.	F1
A	IP	0.988	0.49	0.655
	UA	0.952	0.39	0.554
	IP·UA	0.962	0.603	0.741
	Geo-Core	0.805	0.859	0.831
	Geo-Core·UA	0.808	0.858	0.833
	IP+Geo-Core·UA	0.843	0.83	0.837
B	IP	0.994	0.784	0.877
	UA	0.9	0.521	0.66
	IP·UA	0.937	0.847	0.89
	Geo-Core	0.971	0.968	0.97
	Geo-Core·UA	0.974	0.968	0.971
	IP+Geo-Core·UA	0.975	0.967	0.971

Table 1: Evaluation results for different feature sets

an extension of the AdaBoost technique that combines with bagging [26], and Random Forests (RF) [4]. Each method is provided the same features (IP, user-agent, geo-coordinates) and similarity measures as defined for our PSL models.

Since PSL-models estimate a soft-truth value for the target predicate, SameUser(), we select the optimal cutoff value θ, that best separates positive and negative pairs. For geo-coordinate selection, we consider a single geo-coordinate Geo-Center, which is the center of user geo-coordinates, and Geo-Freq, which is the most frequently occurring geo-coordinate. In addition, we consider the set of all geo-coordinates occurring in each user's previous cookie logs, Geo-All. For fair comparison, we do not use the transitivity rule here, since other classifiers cannot take it as an input. We describe the analysis result of using the transitivity rule with PSL in Section 6.2.

Comparison of feature sets: Table 1 shows the performance of PSL models with different feature sets. Using only IP address, its precision is close to 1 (0.988 and 0.994 on A and B dataset, respectively). However, it achieves a recall of only 0.49, and 0.784. This shows that it cannot resolve cookie logs to the same user, when the user has different IP-addresses. Using user-agent string only, the PSL model achieves even lower recall value while its precision is worse than PSL models with IP. However, by combining it with IP, the performance improves in both, precision and recall. Using Geo-coordinates, with or without additional features, PSL performs generally better than all the other cases achieving F-scores up to 0.837 and 0.971. It indicates that compared to other core features for visitor stitching, Geo-coordinate provides more reliable information when properly processed, as we proposed.

Comparison with other methods: According to the results[4] (Table 2), the other classifiers perform moderately well when the datasets have relatively static feature values. In dataset A, which is closer to the cross-device environment (e.g., a diverse feature values of IP address, UA, and geo-coordinates as shown in Figure 2), PSL model has statistically significant better performance than the other approaches (F-score of 0.837). In dataset B, PSL also performs statistically significantly better than (F-score of 0.971). It shows that probabilistic modeling of uncertainty via PSL is more appropriate to the visitor stitching problem on cross-device web logs.

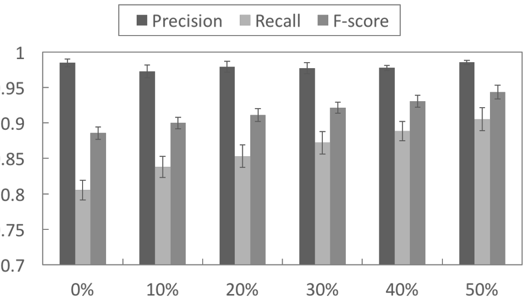


Figure 4: Performance of PSL with different evidence ratio

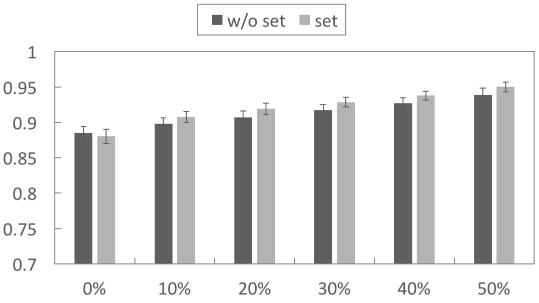


Figure 5: F-score of PSL with instance and set similarity as varying evidence

6.2 Analysis Results

Lack of login information: In many real-world cases, login information is not available and thus we have to handle a set of individual cookies which lack user-level information. PSL is a powerful relational learning technique to leverage relationships between entities via collective inference, which cannot be done with conventional classifiers. To evaluate the effectiveness of PSL in this scenario, we performed additional experiments on logs from dataset B, which contains more multiple-cookie users than A. Specifically, in addition to the rules based on IP, Geo-Core and UA, we add the transitivity rule with varying amounts of SameUser() evidence (0% to 50% compared to the number of pairs in the testing set). Intuitively, if two cookie pairs, (A, B) and (B, C), have a strong signal of SameUser(), we can infer that the cookies A and C are also from the same individual even when A and C do not have enough similarity in terms of features.

According to the results (Figure 4), the performance of PSL increases as more evidence is available with the transitivity rule. As the amount of evidence varies, there are small differences in precision, but the recall always increases with a certain amount of performance gain. It shows that even when we do not have enough user-level information, collective inference through PSL can connect cookies based on relational information and improve the prediction accuracy.

7. CONCLUSION AND FUTURE WORK

We deployed a probabilistic approach based on statistical relational learning to address the visitor stitching problem by resolving individual web logs to unique users. To approach this task, we first introduced the structure of web

[4] The standard errors have been rounded to two decimal places

		Company A			Company B		
Method	Features	Precision	Recall	F-score	Precision	Recall	F-score
PSL	IP+Geo-Center·UA	0.748(.01)	0.847(.01)	0.794(.00)	0.964(.00)	0.941(.00)**	0.952(.00)**
	IP+Geo-Freq·UA	0.725(.01)	0.858(.02)	0.787(.00)**	0.966(.00)*	0.948(.00)**	0.957(.00)
	IP+Geo-Core·UA	0.845(.01)**	0.831(.01)**	**0.837(.00)***	**0.975(.00)**	0.967(.00)	**0.971(.00)****
	IP+Geo-All·UA	0.822(.01)	0.841(.01)	0.831(.00)**	0.973(.00)**	0.958(.00)**	0.966(.00)
NB	IP+Geo-Center·UA	0.936(.00)**	0.652(.01)	0.768(.01)	0.964(.00)*	0.964(.00)	0.964(.00)**
	IP+Geo-Freq·UA	0.928(.00)**	0.651(.01)**	0.765(.01)	0.965(.00)	0.966(.00)	0.965(.00)**
	IP+Geo-Core·UA	0.941(.00)	0.584(.01)	0.72(.01)**	0.962(.00)	0.86(.00)**	0.908(.00)
	IP+Geo-All·UA	**0.945(.00)**	0.592(.01)	0.727(.01)	0.965(.00)	0.857(.00)	0.908(.00)
MB	IP+Geo-Center·UA	0.786(.02)	0.86(.01)	0.818(.00)	0.967(.00)	0.964(.00)	0.966(.00)
	IP+Geo-Freq·UA	0.788(.01)	0.854(.01)	0.819(.00)	0.968(.00)	0.962(.00)	0.965(.00)
	IP+Geo-Core·UA	0.795(.01)	**0.864(.01)**	0.828(.00)	0.945(.00)	**0.97(.00)**	0.957(.00)*
	IP+Geo-All·UA	0.798(.01)	0.859(.01)	0.827(.00)	0.947(.00)**	0.968(.00)	0.958(.00)
RF	IP+Geo-Center·UA	0.819(.01)	0.77(.01)	0.793(.01)	0.964(.00)	0.959(.00)	0.961(.00)**
	IP+Geo-Freq·UA	0.819(.01)	0.771(.01)	0.794(.00)	0.964(.00)*	0.96(.00)	0.962(.00)
	IP+Geo-Core·UA	0.824(.01)	0.769(.01)	0.795(.00)	0.957(.00)**	0.954(.00)	0.956(.00)**
	IP+Geo-All·UA	0.821(.01)	0.767(.01)	0.793(.00)	0.958(.00)	0.954(.00)**	0.956(.00)
LM	IP+Geo-Center·UA	0.885(.00)**	0.761(.01)**	0.818(.00)	0.96(.00)**	0.962(.00)	0.961(.00)**
	IP+Geo-Freq·UA	0.88(.00)**	0.746(.01)**	0.807(.00)**	0.961(.00)	0.964(.00)	0.962(.00)
	IP+Geo-Core·UA	0.877(.01)	0.776(.01)	0.823(.01)	0.949(.00)	0.965(.00)	0.957(.00)
	IP+Geo-All·UA	0.869(.01)**	0.777(.01)	0.82(.01)	0.952(.00)**	0.965(.00)	0.958(.00)**

Table 2: 10-fold cross-validation results for all tested models (Average and standard error); '*' and '' indicate that the entry is statistically significant from the next nearest run on the same set at confidence level of** 95% ($\alpha = 0.05$) **and** 98% ($\alpha = 0.02$)**, respectively**

logs, and discussed the inherent pervasiveness of noisy and incomplete logs. We then highlighted IP addresses, geographic coordinates, and user-agent information generally available features in most web logs, and designed an extensible probabilistic model in Probabilistic Soft Logic (PSL) to utilize these features in a relational learning setting. Finally, we evaluated our PSL models for visitor stitching on two real-world web logs collected via an online marketing solution, and compared them with several state-of-the-art methods. According to the result, our models achieve F-scores of up to 0.837 and 0.971 on two datasets, which are statistically better scores than the state-of-the-art methods. Overall, PSL with the Geo-core feature shows more accurate and balanced performance. In addition, our analysis results show that geo-location has strong predictive signal for visitor stitching. Thus, we propose the method that consistently selects users' core geographic locations as well as embeds it into PSL models with a proper similarity function among the geo-location sets. The performance of the core geographic locations for this problem is intuitive as it can represent people's main locations such as their home or workplace. We also showed how PSL allows collective reasoning to propagate information about user identity. When we are given partial information about user matches, we showed that we can use this to infer remaining user pairs with additional relational information such as transitivity. Overall, our models demonstrate effectiveness in addressing the visitor stitching task in environments where there are many, possibly anonymous, users accessing web services across multiple devices.

Given the massive scale of web log data, one promising area of future work is a distributed inference approach to visitor stitching in PSL. Preliminary explorations of partitioning approaches for PSL models [16, 22] have shown promising results. Combining blocking criteria from entity resolution literature, such as temporal, geospatial or browsing behavior, stitching can be scaled horizontally to accommodate months of web logs across many companies.

8. REFERENCES

[1] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. Hinge-loss markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG], 2015.

[2] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. WWW, pages 357–366, 2008.

[3] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. WWW, pages 61–70, 2010.

[4] L. Breiman and E. Schapire. Random forests. In *Machine Learning*, pages 5–32, 2001.

[5] M. Casado. Peering through the shroud: The effect of edge opacity on ip-based client identification. In *USENIX*, 2007.

[6] D. Coey and M. Bailey. People and cookies: Imperfect treatment assignment in online experiments. WWW, 2016.

[7] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. WWW, pages 761–770, 2009.

[8] A. Dasgupta, M. Gurevich, L. Zhang, B. Tseng, and A. O. Thomas. Overcoming browser cookie churn with clustering. WSDM, pages 83–92, 2012.

[9] P. Eckersley. How unique is your web browser? PETS, pages 1–18, 2010.

[10] S. Fakhraei, B. Huang, L. Raschid, and L. Getoor. Network-based drug-target interaction prediction with probabilistic soft logic. *CBB, IEEE/ACM Transactions on*, 2014.

[11] Google. Google Universal Analytics. `https://developers.google.com/analytics/devguides/collection/analyticsjs/cookies-user-id`, 2015.

[12] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *RecSys*, 2015.

[13] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *PNAS*, 102(33):11623–11628, 2005.

[14] B. London, S. Khamis, S. H. Bach, B. Huang, L. Getoor, and L. Davis. Collective activity detection using hinge-loss Markov random fields. In *CVPR Workshop on Structured Prediction: Tractability, Learning and Inference*, 2013.

[15] A. Malhotra, L. Totti, W. Meira Jr., P. Kumaraguru, and V. Almeida. Studying user footprints in different online social networks. ASONAM, pages 1065–1070, 2012.

[16] H. Miao, X. Liu, B. Huang, and L. Getoor. A hypergraph-partitioned vertex programming approach for large-scale consensus optimization. In *2013 IEEE International Conference on Big Data*, 2013.

[17] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, Aug. 2000.

[18] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd International Workshop on Web Information and Data Management*, WIDM, pages 9–15, 2001.

[19] G. D. Montanez, R. W. White, and X. Huang. Cross-Device Search. CIKM, pages 1669–1678. ACM, 2014.

[20] J. Pujara, B. London, and L. Getoor. Budgeted online collective inference. In *UAI*, 2015.

[21] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Knowledge graph identification. In *ISWC*, 2013.

[22] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Ontology-aware partitioning for knowledge graph identification. In *CIKM Workshop on Automatic Knowledge Base Construction*, 2013.

[23] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi. Linking users across domains with location data: Theory and validation. WWW, pages 707–719, 2016.

[24] R. Saha Roy, R. Sinha, N. Chhaya, and S. Saini. Probabilistic deduplication of anonymous web traffic. WWW Companion, pages 103–104, 2015.

[25] M. Sumner, E. Frank, and M. Hall. Speeding up logistic model tree induction. PKDD, pages 675–683, 2005.

[26] G. I. Webb. Multiboosting: A technique for combining boosting and wagging. In *Machine Learning*, pages 159–196, 2000.

[27] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000.

[28] R. W. White and A. H. Awadallah. Personalizing Search on Shared Devices. In *SIGIR*, 2015.

[29] Y. C. Yang. Web user behavioral profiling for user identification. *Decis. Support Syst.*, 49(3):261–271, 2010.

[30] J. Zhang and P. S. Yu. Integrated anchor and social link predictions across social networks. IJCAI, pages 2125–2131, 2015.