Toward the Attribution of Web Behavior

Myriam Abramson
Naval Research Laboratory
Washington, DC
myriam.abramson@nrl.navy.mil

***Abstract*—As more people browse the Web to gather information, recognizing Web browsing behavior signatures can replace or complement keystroke authentication where *authentication* is defined as the capability of identifying an individual within a set of individuals. We claim that recurring temporal patterns of Web site visits can help identify an individual of interest and, more generally, categorize Web browsing behavior. Furthermore, just like keystroke authentication, attribution of Web behavior is not obtrusive and has applications in cyberwarfare as a new biometric technique. In this paper we describe some exploratory work and preliminary comparative results of machine learning techniques applicable to the attribution of Web browsing behavior problem.**

I. Introduction

Anonymity is difficult to preserve and might not be completely possible. Rather, it is possible to hide behind pseudonymity and unobservability when interacting on the Web. While pseudonimity is the capability to hide behind a false identity, unobservability is the capability to hide in plain sight [1]. Together, pseudonymity and unobservability make a powerful recipe for quasi-anonymity. Consequently, there is a need in cyberspace to know who our attackers are and to track them wherever they might be on the network – a problem known as *attribution*.

Marketers have long been interested in understanding Web interaction behavior [2], [3], [4] in order to design Web sites that entice visitors to finish their Web session with a checkout of their shopping cart. Behavioral targeting is an approach used by advertisers (e.g., DoubleClick) that track Web behavior to deliver advertisements matching an individual's profile. Research in this area has concentrated on identifying the demographic characteristics of a behavior such as age and gender rather than authenticating a single individual [5]. There has also been some research on understanding online browsing behavior from an aggregate perspective in order to identify influential websites in user navigation patterns [6].

Section II reviews the related work in the area of attribution in cyberspace. Section III introduces our proposed methodology in this area with some exploratory evaluation in Section IV. Finally, we conclude in Section V with our direction for research.

II. Related Work

The attribution problem in cyberspace has been addressed in several ways mainly by leveraging from features in the browser (e.g., history stealing, cookies, etc.) or accessing datasets containing partially identifying information. For example, de-anonymization in social networking websites has been accomplished by taking the intersection of users from group memberships in a social network through information from hyperlinks in the browser history and knowledge about those groups [7]. In general, unique identification is possible by cross-referencing independent information sets containing partial information with a universal set in a manner equivalent to a database join (also known as "linkage attacks"). For example, it has been possible to link medical records to individuals in voter registration records [8]. Some success has been reported with the classification of global syntactic features of a Web session (e.g. length of session, average time on a page, etc.) per user [9] but this mode of identification is easy to defeat and only serves as a proof of concept that signature identification is possible with data aggregated over several sessions. It has also been shown that authorship of content can be determined from stylometric features on an internet scale threatening anonymity [10] but this type of attribution depends on published content. Research in predicting user behavior in cyberspace has also been directed toward improving tasks such as information retrieval [11] or desktop assistance [12]. For example, based on the content of the current Web page and a user's original search keywords, the most relevant hyperlinks in the page are highlighted to guide selection of the next page to visit. This type of prediction is oriented toward the information presented in context to the user rather than the specific activity that a user might pursue (e.g. send an email, read a paper, etc.).

In contrast to previous approaches, we address the attribution problem by leveraging both from syntactic patterns in Web browsing history and the semantic content of this history.

III. Proposed Methodology

Our approach to the tracking and authenticating of Web browsing behavior involves the following tasks:

1) Classify Web pages into genres at multiple levels of granularity;
2) Encode temporal sequences of individual Web browsing behavior consisting of Web page genres and pauses from *clickstream* data;
3) Learn Web behavior signatures (profiles) with structured prediction;
4) Recognize an individual or typical behavior or report unknown.

In order to acquire the relevant clickstream data, we have developed a Firefox browser extension to be used in an

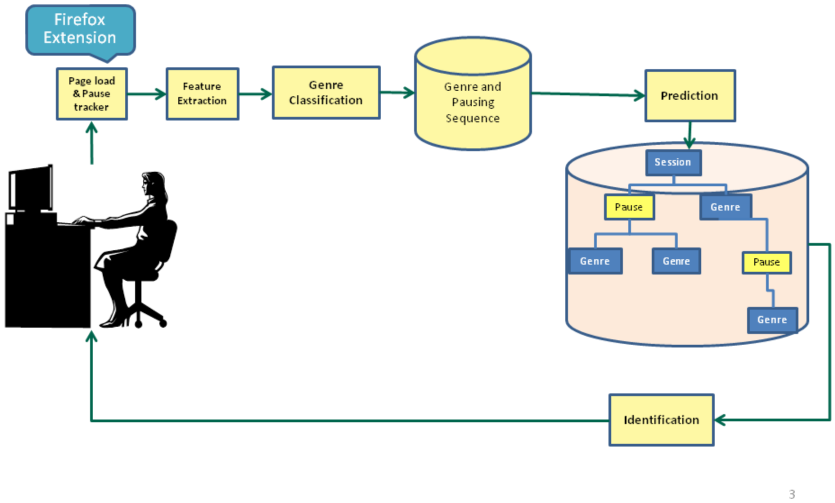

Figure 1. Web behavior attribution data flow

ethnographic user study which has been approved by our institutional review board. Other datasets will also be generated from public social media. The overall dataflow is illustrated in Fig. 1. We claim that genres, as functional categories of information presentation, are more indicative of Web browsing behavior than content alone. For example, it takes longer to read a research paper for certain persons. Identifying Web browsing signatures is a search through abstraction spaces: the varying degree of generality in the semantic content of the Web pages and the syntactic pattern of the behavior (Fig. 2) necessary to uniquely identify someone. To borrow an analogy from natural language processing, taking a set of genres as our lexicon, a Web browsing history as a natural language sentence and a profile as an encoded parse tree, the problem of attribution reduces to the task of identifying the best matching parse tree most suited for a given sentence.

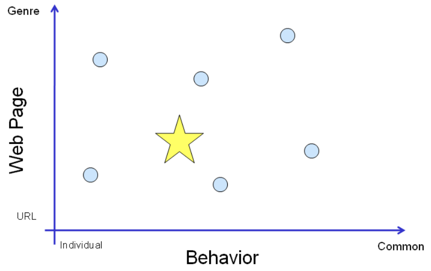

Figure 2. Search through abstraction spaces for Web signatures. Each signature is a dot in the problem space

A. *Structured Prediction*

Structured prediction is a supervised learning method that addresses problems where the output itself is complex [13]. For example, the output can be in the form of a tree, a sequence or a graph. Structured prediction is being activately researched in natural language processing to predict parse trees from sentences and in sequence matching problems such as machine translation [14]. For example, when translating a sentence from English to French, a word for word translation is not enough because it ignores correlations and constraints among words. What has to be predicted here is a set of mappings and not just individual mappings. Because of this selective matching between two sets, the input and output set, structured prediction methods must solve a combinatorial optimization problem. Traditional classification methods make local predictions but (1) what has to be predicted is different from the sum of the parts and (2) the constraints and correlations between the output features themselves can help improve the prediction of the parts. Formally, structured prediction involves learning a mapping from complex inputs $x \in X$ to complex outputs $y \in Y$ from a training sample of input-output pairs (x_i, y_i) drawn from an unknown distribution. There are dependencies between Web page requests that extend beyond the last page visited due to the non-linearity (hypertext) of Web content or inadvertent page clicks.

Hidden Markov Models (HMMs) [15] have long been the traditional method to model behavior from observations but they are limited in their capability to represent constraints between any two states of the output because of their Markov assumption and the independence assumption of the observations. In addition, HMMs become intractable when the observations are not enumerable. Although work has been done on overcoming those limitations resulting in complex models, structured prediction methods provide a unified framework to predict and learn arbitrary activity patterns. In addition, HMMs require enough training data to obtain an accurate generative model of observations while structured prediction methods leverages only from dependencies in the observation sequence and local predictions and therefore might get away with much less data. Structured prediction methods, like maximum entropy Markov models, turn HMMs on their head by conditioning the probability of Y on the observations X making it possible to leverage from modern discriminative classifiers. Structured prediction methods make it possible to solve the HMMs' decoding problem (i.e., finding the output sequence of hidden states from the input sequence of observations) without a complete model capable of generating the observations (i.e., HMMs' learning problem).

B. Approaches to Structured Prediction

There are several approaches to solve the combinatorial problem of structured prediction. They all involve a change of representation that includes tentative predictions of the output sequence. We review below the probabilistic approach of conditional random fields (CRFs) [16] and a reinforcement learning approach [17].

1) CRFs: CRFs are predicated on the interaction of neighboring labels in a sequence. The Ising model, formalized by the theory of random fields, describes how global, emergent properties emerge from local interactions. Similarly, CRFs exploit those local interactions with overlapping features from the observations X and the labels Y, to predict the most likely label sequence conditioned on the observations. The features express correlations and dependencies between the observation sequence and the label sequence, among the observations themselves, or among the adjacent labels themselves. A feature $F_j(\overline{x}, \overline{y})$ is the sum of binary features f_j describing an example $x \in X$ of length n:

$$F_j(\overline{x}, \overline{y}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i) \tag{1}$$

For example, in describing Web page sequences, such a feature f_j for Web page x at position i could be:

$$f_j(y_{i-1}, y_i, x, i) = \begin{cases} 1 & if\, title\, contains\, race \\ & and\, genre = sports\, news \\ & and\, previous\, genre = search\, page \\ 0 & otherwise \end{cases} \tag{2}$$

CRFs overcome the problem of dynamic length sequences by having a fixed set of overlapping features. A discriminative classifier, such as logistic regression, can then be used to learn the weights w_j of those features in a training phase to obtain real-valued features g_i :

$$g_i(y_{i-1}, y_i) = \sum_{j=1}^{J} w_j f_j(y_{i-1}, y_i, x, i) \tag{3}$$

The score $U(n, y_n)$ of the entire label sequence y of length n ending with label y_n can then be computed with the following recurrence relation [18] computed by dynamic programming algorithms such as the Viterbi algorithm (Alg. 1):

$$U(n, y_n) = \max_{y_{n-1}} [U(n-1, y_{n-1}) + g_n(y_{n-1}, y_n)] \tag{4}$$

In our specific problem, genre classifications can still be ambiguous to uniquely associate a Web page genre to an individual. For example, is the feature that a Web page visited is a blog or a political blog important in identifying an individual? CRFs will therefore disambiguate between genre classifications by learning their feature weights while searching for the best sequence of activities to construct an individual profile according to Eq. 4. Figure 3 illustrates the role of features in CRFs.

Algorithm 1 Iterative Viterbi Algorithm for HMMs

```
name: viterbi
input: $M$, % observations
$sprobs$, % state $S$ probabilities
$trans$, % transition probabilities
$eprobs$ % emission probabilities
output:$path$ % most likely state sequence
       $prob$ % path probability
t ←0
foreach $s \in S$
  a[s] = sprobs($s$) * eprobs($m_0$,$s$)
path[t] ← $argmax_s$(a)
t ← t+1
$M$ ←$M \setminus \{m_0\}$
foreach $m \in M$
 foreach $s \in S$
  maxval ←0
  foreach $s' \in S$
    temp ← a[s'] * trans($s'$,$s$)
    maxval ← max(temp,maxval)
  a'[s] ← maxval * eprobs($m$,$s$)
 path[t] ← $argmax_s$(a')
 a ← a'
return path, $\sum_s a[s]$
```

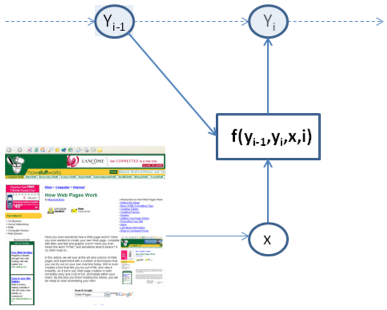

Figure 3. CRF feature linking temporal states y_{i-1} and y_i with document features x.

2) Reinforcement Learning: The identification of a most likely sequence of states was shown to be equivalent to finding an optimal policy for a Markov decision process given a set of states and actions by considering the maximization of expected reward as the minimization of empirical loss on the output training sequence [17]. The actions here are the possible predictions y_i from state $\{x_i, y_{i-1}\}$. Bellman's optimality equation (Eq. 5) is a recurrence relation for sequential decision tasks where $V^*(s)$ is the optimal value of state s, a is the action executed in s that leads to s' such that $\sum_{s'} P^a_{ss'} = 1$ and $P^a_{ss'} = P(s'|s,a)$, $a \in A$, the set of actions, r the current reward, and γ the discount factor, $0 < \gamma \leq 1$, weighting future rewards.

$$V^*(s) = \max_a \sum_{s'} P^a_{ss'} [r^a_{ss'} + \gamma V * (s')] \tag{5}$$

The transition probabilities, $P^a_{ss'}$, can be obtained while training using model-based reinforcement learning [19]. After training, the prediction and evaluation of the most likely sequence for each user can be made by following the optimal policy mapping states to actions learned for this user with Eq. 4 where $g_i(y_{n-1}, y_n) = V^*(s_{n-1})$. In this framework, the stepwise reward has to be proportional to the prediction loss.

Inverse reinforcement learning (IRL) [20] addresses the problem of learning the reward function corresponding to a set of trajectories in the training set such that an optimal policy can be found that will approximate those trajectories. IRL methods can then be applied to structured prediction problems by representing the search through the output space in the prediction problem as a sequential decision making problem [21], [13]. By learning a specific reward function, the optimal policy encapsulates a model of user behavior.

IV. Proof of Concept Evaluation

Before attempting structured induction algorithms, we evaluated several learning algorithms for the Web browsing identification task with the discovery challenge dataset from ECML [22], [23] as proof of concept. In this dataset, a user session is characterized by its timestamp, a sequence of page visited, categorized by page type, and number of page views (page loads) (Table I). For example, the sequence of page type visited given as “12,1 9,3 7,2” corresponds to the chronological sequence “12,9,9,9,7,7”. User sessions were extracted randomly from the training and test sets and the experiments consisted of distinguishing them correctly in the test data based on profiles built from the training data. The training data averaged 89 user sessions ranging from 1-183 page type visited. Four basic types of algorithms to build profiles were compared with random selection based on the class distribution during training as a baseline:

1) Discrete Markov process algorithm based on profiles built from the transition probabilities between page types for each user during training. Each page type observation O_t corresponds to state S_t. The profile with the most likely sequence of transition probabilities between page types was selected for identification.
2) HMMs maximum likelihood training with the Baum-Welch algorithm implemented using Jahmm [24]. Each user profile was built with three fully interconnected hidden states with initial uniform probabilities to model the continuum of a session (beginning, middle, and end). Unlike discrete Markov processes, the page type observation O_t is decoupled from the state S_t in an HMM. The profile with the most likely sequence of hidden states S using the iterative Viterbi algorithm (Alg. 1) was then selected for identification.
3) Classification of users from the frequencies of page types visited with the decision tree algorithm J48 from the Weka machine learning toolbench [25];
4) Classification of users from global syntactic features of the session (number of pages in the session, average number of page views, session length, day of the week and time of day) with the decision tree algorithm J48;

Figure 4 illustrates the comparative results obtained using the weighted F-measure in discriminating between 10 users for each individual Web session. This experiment shows that machine learning techniques are promising in this domain, achieving significant performance results for the first three algorithms taking into account state transitions, state/observation probabilities, and page types while the performance of purely syntactic patterns degrades rapidly. The research will consist of scaling up those results with structured prediction and reinforcement learning and additional information from webpages and clickstream data.

Training examples	Test examples	Users	Page Types	Page Views
380485	166299	4853	1-20	1-117

Table I
ECML/PKDD 2007 Discovery Challenge Dataset

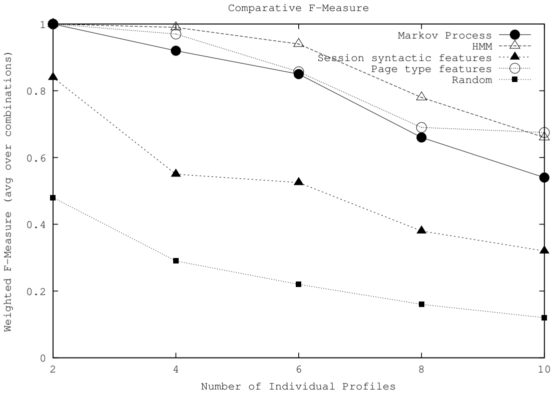

Figure 4. Comparative Results for discrimination between 2-10 users using weighted F-measure

V. Conclusion

We claim that the genre of Web sites visited can tell us something about an individual browser and that how and when the browsing was done is also revealing. For example, the time of day and the length of the pause at a page can give some information. We have shown through simple experiments that identification of users, or distinguishing between users, is possible within a certain accuracy but that performance degrades rapidly as the number of users increases. We claim that structured prediction will allow us to scale up by leveraging additional information in constructing Web behavior signatures. Web browsing has become another dimension of human activity and this methodology could be used for continuous or periodic identification to complement an initial strong identification technique for authentication.

References

[1] E. A. Poe, “The purloined letter,” 1844.

[2] R. W. White and S. M. Drucker, “Investigating behavioral variability in web search,” in *Proceedings of the International World Wide Web Conference WWW07*, 2007.

[3] R. Srikant and Y. Yang, “Mining web logs to improve website organization,” in *Proceedings of the International World Wide Web Conference WWW01*, 2001.

[4] R. Atterer, M. Wnuk, and A. Schmidt, “Knowing the user’s every move: user activity tracking for website usability evaluation and implicit interaction.,” in *Proceedings of the International World Wide Web Conference WWW06*, pp. 203–212, ACM, 2006.

[5] K. De Bock and D. Van den Poel, “Predicting website audience demographics for Web advertising targeting using multi-website clickstream data,” *Fundamenta Informaticae*, vol. 98, no. 1, pp. 49–70, 2010.

[6] R. Kumar and A. Tomkins, “A characterization of online browsing behavior,” in *Proceedings of the 19th international conference on World wide web*, WWW ’10, (New York, NY, USA), pp. 561–570, ACM, 2010.

[7] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, “A practical attack to de-anonymize social network users,” in *IEEE Security and Privacy*, 2010.

[8] L. Sweeney, “Replacing personally-identifying information in medical records, the scrub system.,” in *Proceedings of the AMIA Annual Fall Symposium*, p. 333, American Medical Informatics Association, 1996.

[9] B. Padmanabhan and C. Yang, “Clickprints on the web: Are there signatures in web browsing data?,” tech. rep., Wharton School, University of Pennsylvania, 2006.

[10] A. Narayanan, H. Paskov, N. Gong, J. Bethencourt, E. Stefanov, E. Shin, and D. Song, “On the feasibility of internet-scale author identification,” in *Proceedings of the 33rd Conference on IEEE Symposium on Security and Privacy*, 2012.

[11] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, “Webwatcher: A learning apprentice for the world wide web,” in *AAAI Spring Symposium on Information Gathering from Heterogeneous, distributed environments*, 1995.

[12] T. D. X. Bao, J. Herlocker, “Fewer clicks adn less frustration: reducing the cost of reaching the right folder,” in *International Conference on Intelligent User Interfaces*, 2006.

[13] H. Daume and D. Marcu, “Learning as search optimization: Approximate large margin methods for structured prediction,” in *Proceedings of the 22nd International Conference on Machine Learning ICML*, 2005.

[14] H. Daumé, J. Langford, and D. Marcu, “Search-based structured prediction,” *Machine learning*, vol. 75, no. 3, pp. 297–325, 2009.

[15] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, pp. 257–286, 1989.

[16] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *International Conference on Machine Learning ICML*, 2001.

[17] F. Maes, L. Denoyer, and P. Gallinari, “Structured prediction with reinforcement learning,” *Machine Learning*, 2009.

[18] C. Elkan, “Log-linear models and conditional random fields.” Retrieved from http://cseweb.ucsd.edu/users/elkan/250B/loglinearCRFs.pdf.

[19] C. G. Atkeson and J. C. Santamaria, “A comparison of direct and model-based reinforcement learning,” in *IN INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, pp. 3557–3564, IEEE Press, 1997.

[20] A. Ng and S. Russell, “Algorithms for inverse reinforcement learning,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670, 2000.

[21] G. Neu and C. Szepesvári, “Training parsers by inverse reinforcement learning,” *Machine learning*, vol. 77, no. 2, pp. 303–337, 2009.

[22] “EMCL/PKDD 2007 discovery challenge.” Retrieved from http://www.ecmlpkdd2007.org/.

[23] K. Dembczyński, W. Kotłowski, and M. Sydow, “Effective prediction of web user behaviour with user-level models,” *Fundamenta Informaticae*, vol. 89, no. 2, pp. 189–206, 2008.

[24] J. Francois, “Jahmm: An implementation of hidden markov models in java,” 2010. Software available at http://www.run.montefiore.ulg.ac.be/ francois/software/jahmm/.

[25] M. Hall, E. Frank, G. Holmes, and I. H. W. Bernhard Pfahringer, Peter Reutemann, “The WEKA data mining software: an update,” in *SIGKDD Explorations*, vol. 11, 2009.