Learning Temporal User Profiles of Web Browsing Behavior

Myriam Abramson
Naval Research Laboratory
Washington, DC 20375
myriam.abramson@nrl.navy.mil

Abstract

As more people use the Web to gather information, communicate, work, and otherwise have fun, learning user profiles of Web browsing behavior can help personalize search results and identify persons of interest. Learning individual user profiles from Web browsing behavior also has applications in cybersecurity as a continuous soft biometric technique to verify a user's claim of identity, in online opinion mining to eliminate duplicate users, in fraud detection to discriminate between normal and anomalous behavior, in adaptive user interfaces to contextualize the presentation of information, and in recommender systems based on collaborative filtering techniques. The contribution of this paper is a novel method for modeling Web browsing behavior and learning user profiles using conditional random fields with experimental results in authenticating users from a user study.

1 Introduction

Learning user profiles has many applications and what is learned from modeling a user can vary. For example, an adaptive user interface model might model context in order to help a user in the performance of a task regardless of preferences. Selecting a representation and methodology to learn user profiles depends on the available data and the objective. Cognitive user profiles are generally distinguished from signatures or fingerprinting approaches that can identify a user but cannot make temporal predictions of behavior. In contrast, the power of user models rests in its generality. For example, an opponent model based on a specific player (or self-play) can improve the play of a computer program against other opponents, or a user model for a recommender engine can be leveraged by similar users through collaborative filtering techniques. Debunking cognitive biases and coaching a user to learn new skills (or concepts) are another type of application where group profiles combine with user profiles. In this paper we investigate to what extent user profiles of Web browsing behavior can serve as signatures with experimental results in the authentication task from a user study. Authentication (one-to-one) is a way of evaluating the accuracy of cognitive user profiles without the additional complexity of multi-class classification (one-to-many). Challenges in discovering unique user profiles from Web browsing behavior include the commonality of certain websites across users (e.g., facebook.com) and the difficulty of distinguishing rare but relevant clicks from accidental clicks within a limited training time window.

The paper is organized as follows. In Section 2 we describe prior research in learning user models of Web browsing behavior in different application contexts. In Section 3 we introduce our learning approach based on conditional random fields (CRFs) including our modeling of Web browsing behavior through a user study. To our knowledge, this approach has not been used in the context of Web browsing. In Section 4 we present our empirical study and experimental results for an adapted game of "name that tune" to the authentication decision task. Finally, our conclusions and future work are summarized in Section 5.

2 Related Work

Marketers have long been interested in understanding Web interaction behavior [1] 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 behavior of users with specific demographic characteristics such as age and gender [2] rather than authenticating a single individual or predicting their next click.

In the area of recommender systems and information filtering, learning user profiles from Web users has concentrated on learning relevant categories of webpages based on topic content [3, 4] and user feedback. Work in Web usability has focused on grouping users according to the similarity of their navigational behavior [5] to improve the design of a website. In [6], aggregated user profiles are represented as a collection of weighted URLs and recommendations made according to a matching score with respect to the user active session.

In [7] learning the profiles of cellphones users included features of specific URL domain names extracted from the browser history in conjunction with mobility features. Those user profile features conditioned on time-of-day are combined in a Gaussian Mixture Model to produce an authentication score. Unlike phone numbers, the commonality of certain websites was found to be an issue in detecting mas-

queraders. Learning user profiles from Web browsing behavior has also been approached from a Web usage mining perspective [8] for identification purposes. In this approach, user profiles are constructed from a fixed set of patterns (domain names) occurring together in a Web sessions. Users are then identified with a nearest-neighbor approach or with a classification learning method based on the strength of those patterns. This approach works well in identifying a fixed and limited number of users within a website where the number of possible pages is also limited and fixed.

Modeling individual Web browsing behavior was shown to be useful against distributed denial of service attacks to distinguish abnormal HTTP requests from normal ones [9] on the server side. Web browsing is modeled using a hidden semi-Markov model with explicit duration where the observations are the HTTP requests and the states are the webpages. Because inline HTTP object requests (such as an image logo) overlap among webpages, it is difficult to distinguish user clicks on the server side. A semi-Markov model can model the jump to unrelated webpages through the address bar of the browser or through bookmarks. In this work, normal HTTP requests are distinguished from abnormal ones by thresholding the accepted deviation from the mean entropy of the model. However, this work does not distinguish between normal HTTP requests as belonging to different users.

In [10] stochastic agent-based user models of Web browsing behavior were shown to produce realistic results in the simulation of Web traffic patterns. The user model parameters are based on empirical observations, aggregated over sessions, such as the topical locality of webpages where the temporal distance of clicks is correlated with content similarity. The relevant conclusion of this work was that the broad distribution of aggregated behavior suggests that there is no "typical" user described by their web traffic. However, this work does not attempt to recognize users from their Web browsing behavior.

Path analysis based on Markov models and data mining techniques [11, 12] to reduce model complexity has concentrated on modeling Web usage across users within a website with no unknown webpages. While this approach models common navigation behaviors, it does not model users directly and does not transfer to other websites or across websites.

In contrast to temporal user profiles, authentication from Web browsing behavior has been addressed with a bag-of-words ensemble approach of one-class SVMs [13]. While this approach addresses the discriminative aspect of a signature, temporal user profiles provide added capabilities such as activity recognition and click prediction.

3 Proposed Approach

Hidden Markov models (HMMs) have long been the preferred method to model behavior but they are limited in their capability to represent constraints between any two states because of their Markov assumption and the independence assumption of the observations [14]. Although work has been done to overcome 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 leverage only from dependencies in the observation sequence and local predictions. There are several approaches to solve the combinatorial problem of structured prediction and our approach is based on CRFs [15], a probabilistic approach predicated on the interaction of neighboring label sequences.

To learn user profiles as signatures, it is not enough to predict the activities in a temporal sequence. Rather a sequence of actions should be evaluated given a user model relative to other user models. The probability of a clickstream sequence given by a user model provides confidence toward the attribution of this sequence regardless of the activities recognized by the model. Although temporal graphical models such as CRFs have been primarily used for sequence tagging in natural language parsing [16] and activity recognition [17], another aspect of graphical models is to provide an overall probabilistic evaluation. We first describe our user study to obtain the clickstream data.

3.1 Web Browsing Modeling

Logging of spontaneous clickstream data in this user study consisted of recording through custom-built browser extensions (Firefox and Chrome) the timestamp and the URL that was visible at the time by the user in the address bar of the browser (i.e. *click*). The data was parsed offline to minimize interference with the user. Fourteen subjects (4 females and 10 males) participated in this study during the course of their work for one month. The population was fairly homogeneous and rated themselves highly "Web savvy." Web browsing behavioral data is noisy and requires some pre-processing for analysis. Noise occurs due to distortion from the network behavior, errors in accessing URLs, and automatic page insertion in the browser due to the dynamic reloading of webpages. The clickstream data is parsed into "sessions" where a session is defined as a continuous stream of clicks delimited by pauses greater than 30 minutes as in [18]. User sessions are the data points in our study of Web behavior.

We categorized URLs into page types using Diffbot[1], a Web service for genre classification, that currently categorizes webpages into 21 pages (including "error"). Genres are functional categories of information presentation. In other words, genres are a mixture of style, form, and content. For example, books have many genres such as mystery, science-fiction, fiction, and biography. Similarly, webpages have evolved their own genres (e.g, blog, homepage, article). Basically, the genre of a document is tied to its purpose and reflects social conventions for disseminating and searching information. We claim that genres are more indicative than topics for the attribution of Web browsing behavior. For example, students read tutorials and professors read research papers with high reading complexity regardless of topics. Future work will develop a genre "palette" that enhances the attribution from Web browsing behavior.

[1] http://www.diffbot.com

The following features were extracted from the data: day-of-week, time-of-day, pauses (elapsed time between contiguous clicks), burstiness (rate of change between pauses), and time between revisits within a session. For clarity, we only show the behavior of the same three users in our figures. Figure 1 illustrates three users and their patterns of weekly online activity aggregated across all sessions. Figure 2 illustrates the time-of-day accesses of those users and Fig. 3 illustrates the genre of the webpage accesses. Only those features are used in our user models.

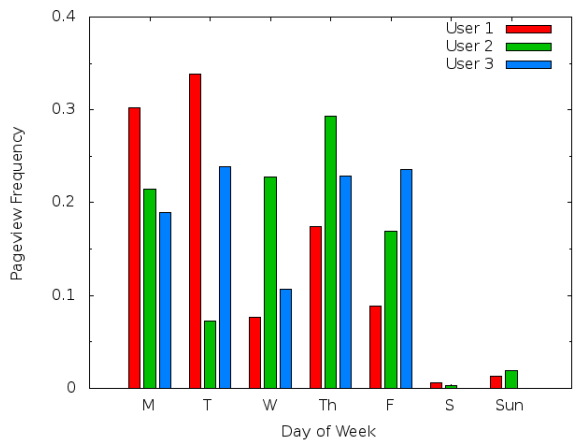

Figure 1: Day-of-week activity patterns for 3 users aggregated across all sessions. There is a lull on Wednesday across users. User 3 is the only one not active during the weekend.

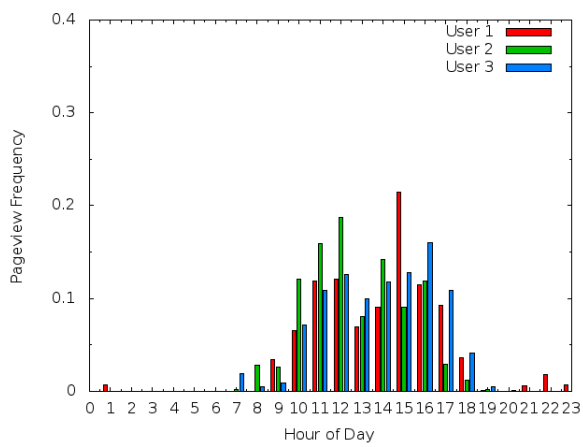

Figure 2: Time-of-day accesses for 3 users aggregated across sessions. There is a lull in the middle of the day across users. User 2 is mostly active in the morning while User 1 is active after dinner.

3.2 Conditional Random Fields

We describe below the derivation of CRFs from basic probabilistic principles and we restrict our discussion to linear-chain CRFs for the prediction of sequences.

The factorization of Bayesian nets according to conditional independence enables the tractable computation of

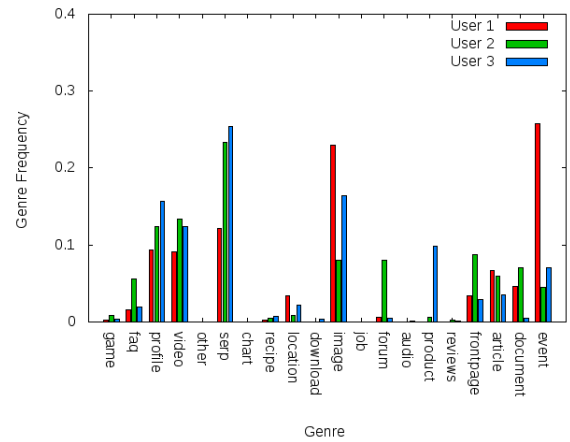

Figure 3: Genre frequencies for 3 users (excluding errors) aggregated across sessions. User 3 is distinguished by the number of "Event" page accesses.

the joint probability of a collection of random variables $P(\bar{y})$ according to the structure of a graphical model as follows.

$$P(\bar{y}) = \prod_i p(y_i|y_i^p) \tag{1}$$

where y_i^p are the parents of y_i. However, it is sometimes more natural to model a problem according to spatial or temporal proximity of the nodes rather than their conditional independence. For example, in a lattice-like graphical structure, the Markov blanket of a node does not obey the spatial neighborhood properties of the graph as expected [19]. It is therefore more natural to model such graphs as undirected graph models where the independence of the nodes is determined only by the absence of a connecting edge. It is possible to convert a directed graph to an undirected graph by "moralizing" it (i.e. adding edges between nodes to indicate implicit dependence). The edges of an undirected graph cannot be weighted by conditional probabilities anymore but can be evaluated according to the "affinity" of the nodes defined by a *potential* function or *factor* $\varphi(x, y)$. More generally, a factor φ associates a set of related variables y_c to a value $\mathbb{R}$. Those factors are parameterized by a weight θ_c that can be learned from data using various methods. According to the Hammersley-Clifford theorem, the joint probability of the graph can then be obtained as follows:

$$P(\bar{y}|\bar{\theta}) = \frac{1}{Z(\theta)} \prod_c \varphi(y_c|\theta_c) \tag{2}$$

where $Z(\theta)$ is the partition function normalizing the product of factors in order to obtain a probability distribution.

CRFs leverage from the undirected graph modeling approach to model the conditional distribution $P(\bar{y}|\bar{x})$ of a set of target variables $\bar{y}$ and a set of observed variables $\bar{x}$ to represent structured data. A factor represents the probability of a target variable y as a linear function of the weight parameters θ and the observed input variables $\bar{x}$ which are not necessarily independent. If the observed input variables are indicator functions, $\phi(x, y)$, then $P(y|\bar{x}, \bar{\theta})$ is defined as follows:

$$P(y|\bar{x},\bar{\theta}) = \frac{\exp \sum_j \theta_j \phi_j(\bar{x},y)}{\sum_{y' \in Y} Z(\bar{x},y')} \quad (3)$$

where $Z(\bar{x},y') = \exp \sum_j \theta_j \phi_j(\bar{x},y')$. The weight parameters θ of the factors are learned using discriminative learning methods for the target variable y (see 3.3). There are two types of feature functions in representing a sequence [20]: (1) edge functions between two labels and (2) pairwise functions relating x and y. Generally, $f_j(y_{t-1}, y_t, \bar{x}, t)$ represents a feature function (Fig. 4). CRFs are usually distinguished from maximum-entropy approaches [21] by conditioning over the entire observation sequence. Generalizing to "global" factors over the entire sequence of observations where $F_j(\bar{x},\bar{y}) = \sum_t^n f_j(y_{t-1}, y_t, \bar{x}, t)$ and where t is the position in the sequence of length n, the probability of the sequence $\bar{y}$ is then:

$$P(\bar{y}|\bar{x},\bar{\theta}) = \frac{1}{Z(\bar{x},\bar{\theta})} \exp(\sum_j \theta_j F_j(\bar{x},\bar{y})) \quad (4)$$

where $Z(\bar{x},\bar{\theta}) = \sum_{\bar{y} \in nP_Y} \exp(\sum_j \theta_j F_j(\bar{x},\bar{y}))$. The normalization constant Z is computationally intractable but does not need to be computed when predicting the labels in the sequence given the weights of the feature functions. However, when comparing different predictions from different models with possibly different training data, Z needs to be computed in order to comparatively evaluate the probability of a sequence given a model relative to other models. An algorithm to compute Z in linear-chair CRFs is the all-pairs shortest path algorithm based on fast matrix multiplication which is $O(m^3)$ where m is the cardinality of the label set Y [15].

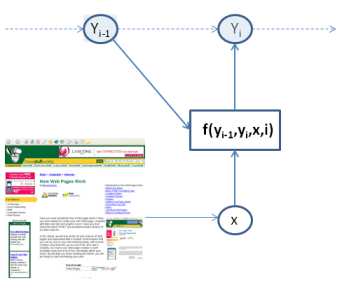

Figure 4: CRFs feature function relating observations and labels.

In our approach, the Viterbi algorithm [14, 16] guides the step-by-step predictions to maximize the choice of each label y in the sequence albeit adding to the computational complexity of the overall algorithm. $P(\bar{y}|\bar{x},\bar{\theta})$ is then defined as follows:

$$P(\bar{y}|\bar{x},\bar{\theta}) = \frac{\arg\max_{t_1 \ldots t_n} \prod_{t=1}^{n} P(y_t|y_{t-1}, \bar{x}_t, \bar{\theta})}{Z(\bar{x},\bar{\theta})} \quad (5)$$

Algorithm 1 describes the Viterbi evaluation of a sequence of observations $\bar{x}$ delimited by START and STOP tags combined with the probabilistic evaluation of CRFs. This algorithm is used in the evaluation of two different discriminative feature weight learning methods described below (3.3). The time complexity of this algorithm is $O(nm^3)$ due to the computation of Z. The learned feature function weights are normalized to have zero mean and unit variance. In addition, very small weights (< 0.0001) are set to zero for regularization purposes. Following [15], matrix $M_i(y,y'|x_i)$ collects the scores for all label transitions (y,y') at each step i in the sequence and Z is the START-STOP entry of the product of these matrices.

Algorithm 1 Viterbi algorithm for CRFs where forward and backtrack are functions as in the Viterbi algorithm, get_matrix collects the scores for all label transitions (y,y') at step t, get_score evaluates a sequence of nodes by multiplying the corresponding entries in the matrices M, and partition_score calculates the normalization constant Z as the START-STOP entry of the product of the matrices.

input: model, $\bar{x}$ //model and observation sequence
output: $\bar{y}$, $P(\bar{y}|\bar{x},\bar{\theta})$ //label sequence and probability
viterbi_crfs (model,$\bar{x}$)=
  //normalize weights
  model ← scale_weights (model)
  t ← 0
  $M_0(y_0, START|x_0)$ ← get_matrix (t,model,$\bar{x}$)
  history_m[t]← $M_0(y_0, START|x_0)$
  history_a[t] ← forward (history_m)
  while t < length ($\bar{x}$)
    t ←t + 1
    $M_t(y_t, y_{t-1}|x_t)$ ← get_matrix (t,model,$\bar{x}$)
    history_m[t] ← $M_t(y_t, y_{t-1}|x_t)$
    history_a[t] ← forward (history_m)
  end
  $\bar{y}$ ← backtrack (history_a, STOP)
  score ← get_score ($\bar{y}$, history_m)
  Z ← partition_score (history_m)
  $P(\bar{y}|\bar{x},\bar{\theta}) \leftarrow \frac{score}{Z}$
return $\bar{y}$, $P(\bar{y}|\bar{x},\bar{\theta})$

3.3 Feature Weight Learning

Several discriminative learning methods can be used to learn the parameters (i.e. the weight of the feature functions). We provide experimental results for three learning approaches.

Label maximum likelihood (hereafter LMK) is a parameter optimization method found in the sequence tagging tool provided by Mallet [22] which consists of maximizing the conditional log likelihood of the labels over the entire training set using limited-memory BFGS optimization. Taking the default parameters, this optimization runs until convergence up to 500 iterations.

The perceptron algorithm for CRFs [16] (hereafter PC-TRN/VTRB) is an iterative algorithm for modifying the weights of the feature functions depending on whether the label, as predicted by the Viterbi algorithm, was correct (Alg. 2). In our implementation, this algorithm runs for 100 iterations on misclassified sequences in the training set

or until no misclassified labels can be found. A mini-batch approach was used where the prediction error was averaged over 5 examples. The weights of the feature functions are initialized with random zero-mean values.

Algorithm 2 Feature weight update for PCTRN/VTRB where y is the label, $\hat{y}$ is the predicted label by the Viterbi algorithm, $\bar{x}$ is the sequence of clicks for one session, edge-ffns are the edge-observation feature functions and node-ffns are the node-observations feature functions.

```
update_weight (model,y_t,ŷ_t,x̄,y_{t-1},ŷ_{t-1})=
 if (y_t ≠ ŷ_t)
  for edge-ffn ∈ model.edge-ffns
   edge-ffn.weight ←edge-ffn.weight
                    + edge-ffn (x̄,y_t,y_{t-1})
                    - edge-ffn (x̄,ŷ_t,ŷ_{t-1})

  for node-ffn ∈ model.node-ffns
   node-ffn.weight ←node-ffn.weight
                    + node-ffn (x̄,y_t,y_{t-1})
                    - node-ffn (x̄,ŷ_t,ŷ_{t-1})
 endif
 return model //updated model
```

Stochastic gradient (ascent in this case) is also an iterative procedure that often outperforms standard optimization algorithms. That algorithm modifies the weights one example at a time along the gradient. In our implementation, the gradient for the weight θ_j of each feature function j is calculated as the difference between empirical counts and expected counts with L2 regularization as follows where n is the length of the sequence:

$$\nabla\theta_j = \sum_{t=1}^{n} \phi_j(x_t, y_t) - \sum_{t=1}^{n} \sum_{y' \in Y} \phi_j(x_t, y') P(y'|x_t, \theta_j) - 2\lambda\theta_j \tag{6}$$

Our stochastic gradient ascent implementation (hereafter SG/VTRB) initializes the weights of the feature functions with random zero-mean values and runs for 100 iterations on randomly accessed training examples. Here too, the weights of the feature functions are initialized with random zero-mean values. The learning rate $\frac{1}{t}$ progressively decays as a function of the iteration number t while λ, the regularization parameter, stays fixed at 0.09.

3.4 Model Representation

We model Web browsing behavior with a linear-chain CRFs and a one-to-one mapping between clicks and genres. The node observations in our model are extracted from click-stream data as follows: (1) the URL for each click is decomposed into ngrams delimited by punctuation characters and overlapping 5 characters; the correlation of ngrams with genres is evaluated using mutual information and the 500 most-correlated ngrams from the training set are kept for each user model (Table 1 shows the 5 most-correlated ngrams found per user in the training set). Previously, [23] has shown the feasibility of this approach for genre classification. Future work will reduce the cross-correlation of ngrams within a URL; (2) the time-of-day; and (3) the day-of-week. The labels (i.e. hidden nodes) in the model are the genres as provided by a genre classifier (Diffbot) during training in a weakly-supervised manner. Consequently, the profile constructed is also a personalized classifier of genres. Pairwise feature functions are generated to relate each observation (i.e. click) to its corresponding label (i.e. genre). To avoid exponential complexity in the number of features, the feature functions are generated strictly from the training data, including the edge-observation features. The number of features functions in our user models ranges from ~700 to ~1000 including a bias feature function. Examples of the feature functions in our model are as follows:

$$\phi_j(y_{i-1}, y_i) = \begin{cases} 1 & if\ y_i = serp\ and \\ & y_{i-1} = profile \\ 0 & otherwise \end{cases}$$

$$\phi_j(y_i, x_i) = \begin{cases} 1 & if\ url(x_i)\ contains\ "googl"\ and \\ & y_i = serp \\ 0 & otherwise \end{cases}$$

$$\phi_j(y_i, x_i) = \begin{cases} 1 & if\ dow(x_i) = Th\ and \\ & y_i = event \\ 0 & otherwise \end{cases}$$

$$\phi_j(y_i, x_i) = \begin{cases} 1 & if\ tod(x_i) = 11\ and \\ & y_i = product \\ 0 & otherwise \end{cases}$$

where x_i represents a click in the sequence.

Since it is not always possible to categorize the genre of a webpage during training (due to connection errors or page obsolescence), we substituted missing genre values (approximately 5% of the data) with the stochastic imputation of genres according to transition probabilities in the user dataset. Table 2 illustrates the data characteristics in our empirical study.

Table 1: 5-most-correlated ngrams with genres (using mutual information) per user in training.

User	ngrams
1	navy, nrl, mil, googl, oogle
2	googl, oogle, ecamp, secam, aseca
3	q, nrl, navy, duckd, duckg
4	mil, navy, aq, 8, utf
5	mil, navy, q, nrl, sear
6	azett, zette, gazet, mca, marin
7	file, hrome, chrom, url, j
8	app, webhp, earch, searc, ource
9	mail, u, inbox, calen, alend
10	nrl, mil, navy, aq, zilla
11	earch, ource, sourc, searc, 8
12	mil, navy, eline, pelin, nrl
13	navy, nrl, mil, google, oogle
14	mail, ca, u, inbox, ender

Table 2: User dataset characteristics where the number of feature functions were extracted from the training sets of one run.

User	#clicks	#sessions	#sessions ≥5 clicks	#feature functions
1	1817	99	61	738
2	2244	121	93	624
3	1899	81	58	737
4	6158	182	148	917
5	6231	206	161	833
6	1299	40	32	641
7	1864	60	36	703
8	992	46	36	653
9	6210	125	99	756
10	2514	129	95	729
11	5716	134	106	824
12	11561	95	83	566
13	7363	81	66	648
14	3577	100	73	635

4 Empirical Study

Can you authenticate that user in 5 clicks? Our experimental hypothesis is that the accuracy of temporal user profiles of Web browsing behavior learned from clickstream data with CRFs is competitive with bag-of-words (BoW) user models. While the aggregation of data found in BoW models reduces noise, temporal user profiles are prone to overfitting over long sequences of clicks. Consequently, learning over the first 5 clicks might better evaluate our hypothesis. Toward that end, we conducted the following experiments using the data obtained from our user study. The user datasets are partitioned equally into a training and test set preserving temporal order and the first 5 clicks of each session are kept. We compare the performance of different algorithms for learning the weights of the feature functions in linear-chain CRFs with identical feature functions using the false rejection rate (FRR) and false acceptance rate (FAR) metrics of authentication. A lower rate for both of these metrics indicates a higher accuracy. The FRR is calculated by applying the user model learned from the training set to the user's own test set. A complete model is obtained by additional learning from the test set. The FAR is calculated by applying the complete model to the test set of all the other users combined. A threshold determined by the equal error rate (EER) (i.e. the threshold minimizing the difference between FRR and FAR) indicates whether a user was authenticated.

We compare the results with the BoW approach of one-class SVM (OCSVM) using the genre frequencies and the genre bigrams frequencies contained in the first 5 clicks, the day-of-week, and the time-of-day as features where the genre has been determined by the Diffbot genre classifier. We used the OCSVM available with LibSVM [24]. Similarly to the temporal models, the FRR is calculated by applying the user model learned from the training set to the test set and the FAR is calculated by applying a complete user model to the test set of all the other users combined. Unlike temporal CRFs models, the FRR is determined by the unclassified instances on the user test set and the FAR is determined by the classified instances on the test set of all the other users combined.

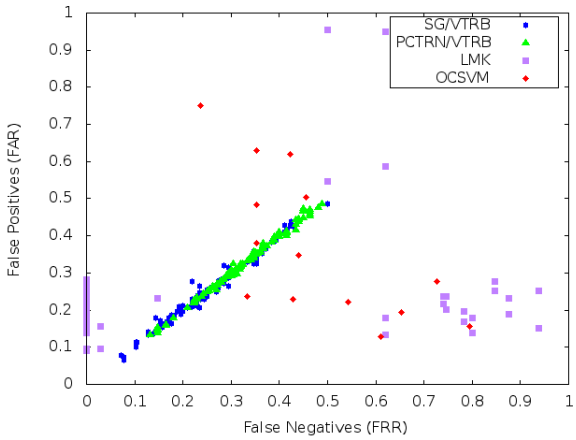

Figure 5: Comparative evaluation (each point represents the EER of one run for one user determined by a threshold minimizing the difference between the FRR and FAR)

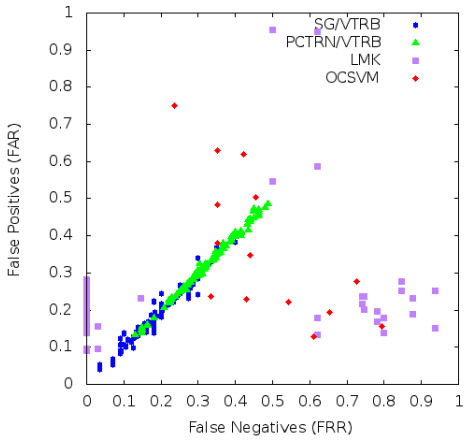

Figure 6: Comparative evaluation (each point represents the EER of one run for one user determined by a threshold minimizing the difference between the FRR and FAR)

Table 3 shows the FRR and FAR results for each user averaged over 10 runs. Figure 5 illustrates the same results in a graphical representation where each point denotes the FRR and FAR results of one run for one user. Since FRR and FAR results are related, we consider only the FRR results in our significance tests. While authentication outcomes vary widely across users in the case of LMK, all the CRFs algorithms perform significantly better than the BoW approach in OCSVM (two-sided t-test p-value < 0.0001). SG/VTRB performs significantly better than LMK and PCTRN/VTRB (two-sided and one-sided t-test p-value < 0.0001). There is no significant difference between LMK and PCTRN/VTRB (two-sided t-test p-value 0.09). Experimental results with a hidden Markov model (HMM/VTRB) using combinations of day-of-week

Table 3: Results per user obtained from the different CRFs weight update algorithms and OCSVM. "Best" results per user ($\min\max(FRR, FAR)$) are highlighted.

User	**OCSVM**		**LMK**		**SG/VTRB**		**PCTRN/VTRB**	
	FRR(%)	FAR(%)	FRR(%)	FAR(%)	FRR(%)	FAR(%)	FRR(%)	FAR(%)
1	0.42±0.00	0.61±0.00	0.71±0.09	0.27±0.24	**0.21**±0.03	**0.22**±0.03	0.34±0.07	0.34±0.06
2	0.79±0.00	0.16±0.00	0.00±0.00	0.24±0.00	**0.13**±0.05	**0.13**±0.06	0.15±0.01	0.15±0.01
3	0.65±0.00	0.19±0.00	**0.01**±0.04	**0.21**±0.02	0.35±0.06	0.35±0.06	0.44±0.03	0.44±0.03
4	0.24±0.00	0.75±0.00	0.84±0.00	0.25±0.01	**0.25**±0.04	**0.24**±0.04	0.33±0.03	0.32±0.03
5	0.45±0.00	0.50±0.00	**0.01**±0.01	**0.15**±0.03	0.29±0.13	0.25±0.04	0.25±0.03	0.25±0.03
6	0.43±0.00	0.23±0.00	**0.00**±0.00	**0.22**±0.03	0.29±0.17	0.29±0.12	0.43±0.04	0.43±0.04
7	0.35±0.00	0.48±0.00	0.47±0.25	0.65±0.39	**0.30**±0.08	**0.29**±0.06	0.39±0.03	0.39±0.03
8	0.35±0.00	0.38±0.00	0.78±0.00	0.19±0.01	**0.26**±0.09	**0.30**±0.09	0.33±0.05	0.33±0.06
9	0.33±0.00	0.24±0.00	0.50±0.37	0.22±0.02	**0.22**±0.05	**0.25**±0.09	0.26±0.03	0.26±0.03
10	0.44±0.00	0.35±0.00	0.87±0.00	0.23±0.01	**0.22**±0.07	**0.22**±0.05	0.30±0.02	0.30±0.02
11	0.35±0.00	0.63±0.00	**0.00**±0.00	**0.19**±0.01	0.31±0.05	0.22±0.07	0.27±0.03	0.27±0.03
12	0.61±0.00	0.13±0.00	**0.23**±0.42	**0.24**±0.03	0.31±0.05	0.31±0.05	0.25±0.03	0.25±0.02
13	0.73±0.00	0.28±0.00	**0.00**±0.00	**0.14**±0.04	0.26±0.09	0.31±0.05	0.31±0.02	0.43±0.02
14	0.54±0.00	0.22±0.00	0.74±0.00	0.23±0.00	**0.18**±0.09	**0.18**±0.03	0.24±0.07	0.31±0.02
Avg.	0.47±0.16	0.37±0.19	0.37±0.39	0.24±0.16	**0.26**±0.09	**0.26**±0.09	0.32±0.02	0.32±0.08

Table 4: Results per user obtained from the different CRFs weight update algorithms and OCSVM. "Best" results per user ($\min\max(FRR, FAR)$) are highlighted.

User	**OCSVM**		**LMK**		**SG/VTRB**		**PCTRN/VTRB**	
	FRR(%)	FAR(%)	FRR(%)	FAR(%)	FRR(%)	FAR(%)	FRR(%)	FAR(%)
1	0.53±0.00	0.23±0.00	0.71±0.09	0.27±0.24	**0.11**±0.02	**0.11**±0.02	0.34±0.07	0.34±0.06
2	0.75±0.00	0.03±0.00	0.00±0.00	0.24±0.00	**0.05**±0.02	**0.05**±0.01	0.15±0.01	0.15±0.01
3	0.55±0.00	0.28±0.00	**0.01**±0.04	**0.21**±0.02	0.27±0.03	0.26 ±0.03	0.44±0.03	0.44±0.03
4	0.60±0.00	0.35±0.00	0.84±0.00	0.25±0.01	**0.15**±0.04	**0.15**±0.04	0.33±0.03	0.32±0.03
5	0.73±0.00	0.33±0.00	**0.01**±0.01	**0.15**±0.03	0.21±0.03	0.21±0.02	0.25±0.03	0.25±0.03
6	0.80±0.00	0.10±0.00	**0.00**±0.00	**0.22**±0.03	0.25±0.09	0.26±0.08	0.43±0.04	0.43±0.04
7	0.72±0.00	0.16±0.00	0.47±0.25	0.65±0.39	**0.20**±0.04	**0.19**±0.04	0.39±0.03	0.39±0.03
8	0.63±0.00	0.23±0.00	0.78±0.00	0.19±0.01	**0.17**±0.05	**0.17**±0.05	0.33±0.05	0.33±0.06
9	0.37±0.00	0.11±0.00	0.50±0.37	0.22±0.02	**0.20**±0.02	**0.20**±0.02	0.26±0.03	0.26±0.03
10	0.55±0.00	0.15±0.00	0.87±0.00	0.23±0.01	**0.16**±0.03	**0.16**±0.03	0.30±0.02	0.30±0.02
11	0.56±0.00	0.39±0.00	0.00±0.00	0.19±0.01	**0.13**±0.04	**0.13**±0.04	0.27±0.03	0.27±0.03
12	0.44±0.00	0.13±0.00	0.23±0.42	0.24±0.03	**0.15**±0.03	**0.14**±0.03	0.25±0.03	0.25±0.02
13	0.80±0.00	0.15±0.00	**0.00**±0.00	**0.14**±0.04	0.23±0.07	0.24±0.07	0.31±0.02	0.43±0.02
14	0.82±0.00	0.05±0.00	0.74±0.00	0.23±0.00	**0.11**±0.02	**0.11** ±0.02	0.24±0.07	0.31±0.02
Avg.	0.63±0.14	0.32±0.45	0.37±0.39	0.24±0.16	**0.17**±0.07	**0.17**±0.07	0.32±0.02	0.32±0.08

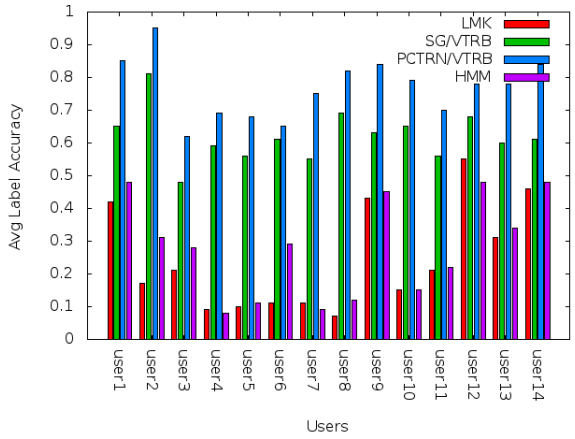

Figure 7: Average label accuracy comparison

and time-of-day as observations and genres as states did not yield meaningful results in the authentication task.

The average label accuracies obtained from the different CRFs algorithms are illustrated in Fig. 7. We also compare the results with HMM/VTRB. Unlike structured prediction methods, OCSVM does not predict the sequence itself so no label accuracies can be provided by this method. There is a significant difference between the label accuracies provided by SG/VTRB and PCTRN/VTRB versus LMK (two-sided t-test p-value < 0.0001). The strong bias of LMK results in predicting the most common labels (genres) thereby reducing the overall label accuracy of this method. There is no significant difference between the label accuracies of LMK and HMM/VTRB. PCTRN/VTRB performs significantly better than SG/VTRB (two-sided and one-sided t-test p-value < 0.0001) which is not surprising since it directly trains from the prediction error rather than bootstrap from the expected counts (Eq. 6).

5 Conclusion

This paper has shown that temporal order matters when learning short user profiles of Web browsing behavior. In this case, we have shown that the accuracy of temporal CRFs models based on clickstream data only is competitive with BoW learning methods such as OCSVM using high-level features (i.e. genres). Consequently, methods discovering duplicate or similar users will gain by including temporal information as context. We have also shown that CRFs, using the Viterbi algorithm, have high label accuracy useful for activity recognition. Future work will investigate how to evaluate longer sequences of data with sliding windows. CRFs offers a flexible way of modeling temporal user profiles and future work in the personalization of Web browsing will include automatic tuning of parameters and finding discriminating feature functions. We will also examine the contribution of relational features (i.e. inter-node observations such as pauses, revisits, etc.) in the performance accuracy of the learned temporal models.

References

[1] 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.

[2] 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.

[3] E. Bloedorn, I. Mani, and T. R. MacMillan, "Representational issues in machine learning of user profiles," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference*, 1996.

[4] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine learning*, vol. 27, no. 3, pp. 313–331, 1997.

[5] G. Paliouras, C. Papatheodorou, V. Karkaletsis, P. Tzitziras, and C. D. Spyropoulos, "Learning communities of the ACAI 99 web site visitors," in *Proceedings of the ACAI 99 Workshop on Machine learning in user modeling*, pp. 65–73, 1999.

[6] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on web usage mining," *Communications of the ACM*, vol. 43, no. 8, pp. 142–151, 2000.

[7] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *Information Security*, pp. 99–113, Springer, 2011.

[8] Y. C. Yang, "Web user behavioral profiling for user identification," *Decision Support Systems*, vol. 49, no. 3, pp. 261–271, 2010.

[9] Y. Xie and S.-Z. Yu, "A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 1, pp. 54–65, 2009.

[10] M. R. Meiss, B. Gonçalves, J. J. Ramasco, A. Flammini, and F. Menczer, "Modeling traffic on the web graph," in *Algorithms and Models for the Web-Graph*, pp. 50–61, Springer, 2010.

[11] M. A. Awad and I. Khalil, "Prediction of user's web-browsing behavior: Application of markov model," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1131–1142, 2012.

[12] J. Pitkow and P. Pirolli, "Mining longest repeating subsequences to predict WWW surfing," in *Proc. USENIX Symp. On Internet Technologies and Systems*, p. 1, 1999.

[13] M. Abramson and D. W. Aha, "User authentication from web browsing behavior," in *Florida Artificial Intelligence Society FLAIRS-26*, 2013.

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

[15] 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.

[16] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 1–8, Association for Computational Linguistics, 2002.

[17] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, p. 235, ACM, 2007.

[18] 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.

[19] K. P. Murphy, *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[20] C. Sutton and A. McCallum, "An introduction to conditional random fields," *arXiv preprint arXiv:1011.4088*, 2010.

[21] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, vol. 951, pp. 591–598, 2000.

[22] "Machine learning for language toolkit." Retrieved from http://mallet.cs.umass.edu/sequences.php.

[23] M. Abramson and D. Aha, "What's in a url? genre clasification from urls," in *AAAI Workshop on Intelligent Techniques for Web Personalization and Recommendation ITWP*, 2013.

[24] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, no. 3, pp. 582–588, 2000.