Hidden Conditional Random Fields with Distributed User Embeddings for Ad Targeting

Nemanja Djuric, Vladan Radosavljevic, Mihajlo Grbovic, Narayan Bhamidipati
Yahoo Labs
701 First Avenue, Sunnyvale, CA 94089, USA
e-mail: {nemanja, vladan, mihajlo, narayanb}@yahoo-inc.com

Abstract—**Estimating a user's propensity to click on a display ad or purchase a particular item is a critical task in targeted advertising, a burgeoning online industry worth billions of dollars. Better and more accurate estimation methods result in improved online user experience, as only relevant and interesting ads are shown, and may also lead to large benefits for advertisers, as targeted users are more likely to click or make a purchase. In this paper we address this important problem, and propose an approach for improved estimation of ad click or conversion probability based on a sequence of user's online actions, modeled using Hidden Conditional Random Fields (HCRF) model. In addition, in order to address the sparsity issue at the input side of the HCRF model, we propose to learn distributed, low-dimensional representations of user actions through a directed skip-gram, a neural architecture suitable for sequential data. Experimental results on a real-world data set comprising thousands of user sessions collected at Yahoo servers clearly indicate the benefits and the potential of the proposed approach, which outperformed competing state-of-the-art algorithms and obtained significant improvements in terms of retrieval measures.**

I. Introduction

Over the previous decade, income generated by the leading internet companies through online advertising has been growing steadily at the amazing rates, with the total annual revenue reaching tens of billions of dollars in the US alone [1]. This burgeoning, highly-competitive, multi-billion industry consists of several key players: 1) advertisers, companies that want to promote their products or services (e.g., Nike, Travelocity) and seek to maximize the response to their advertising campaigns in terms of number of clicks or purchases (referred to as conversions); 2) publishers, websites that host the advertisements and choose when and to whom to show them (such as Google, Yahoo, or Facebook); and 3) online users that are being targeted by the publishers. As both publishers and advertisers work on limited budgets (in terms of monetary funds and available ad space, respectively), focusing the advertising campaigns only on a subset of users who are most likely to click or convert helps attain high response rates while reducing costs to both advertisers and publishers (in terms of poor online experience to their users and missed opportunities). This task is commonly referred to as targeted advertising.

Typically, the publishers offer several types of contracts to the advertisers when setting up the advertising campaigns. In the cost-per-click (CPC) pricing model the advertisers pay only when users click on their ads. Another common pricing model is the cost-per-mille (CPM) model, where the advertisers pay for a certain number of ad impression regardless of the subsequent number of user clicks, commonly used in brand awareness campaigns. Under the both pricing models, success of an online campaign is often quantified through a click-through rate (CTR), defined as a number of ad clicks per hundred ad impressions [2]. Popularity of CTR towards measuring how well the campaign performs is due to several reasons: (a) clicks are a reasonable proxy to user interest and engagement that advertisers can reliably log and measure [3]; and (b) an ad click, unlike a hover or a follow-up search, takes the user directly to advertiser's landing page and thereby closer to the user action that the advertiser wants. Therefore, identifying when and which ads would interest a user enough to entice a click can help publishers maximize their revenue, while at the same time improving online experience for the users as only interesting and relevant ads are displayed.

Due to these reasons and many open research questions related to the large scale and the complexity of targeted advertising problems, the task of increasing the CTR measure through better matching of users with relevant ads has received a lot of attention by the data mining and machine learning communities [4], [5]. Most existing algorithms formulate the problem as a classification or a ranking task, and learn a model based on users' declared or inferred demographic information, their behavioral history (e.g., pages visited, issued queries), meta information of the ads, as well as the context information such as timestamp, publisher, or search queries in the case of sponsored search ads. Then, for each online user, the trained model is evaluated on a pool of available ads and the ones that result in the highest matching score are displayed.

However, using the information aggregated over large periods of time or using only the immediate context information such as current search query might be suboptimal, as more delicate, short-term signals that are useful for estimation of ad click probability may be lost [6]. In particular, actual sequence of user actions can be a strong indicator of their ad click propensity (e.g., user-generated query ”toyota corolla” followed by the query ”front headlight” indicates that the user is much more likely to make an auto part purchase at that very moment than query ”toyota corolla” followed by ”speed racing”), which models that do not explicitly consider a short-term sequence of user actions (referred to as sessions) might not detect. Then, once such a sequence of actions that signals that the user is very likely to click on an ad is found, we can use this information to improve the ad targeting performance. In particular, by detecting the user intent in such a manner, we increase the value, and consequently the revenue, of each shown ad (as the ads that are more likely to be clicked on will be shown to users) and can also reduce the number of shown ads (i.e., by not showing ads to non-interested users, thus directly improving user online experience).

To this end, we explore the benefits of modeling user online sessions using Hidden Conditional Random Fields (HCRF) [7], a powerful and very flexible supervised sequence modeling method, to predict when a user is in the ”click” mode. Moreover, we address a common problem in the ad click prediction tasks: sparsity of the input space due to large cardinality of a set of possible user actions. To address this issue, we propose to learn distributed low-dimensional representations of user actions using directed skip-gram model, an extension of recently proposed *word2vec* learning framework [8]. We compared a number of state-of-the-art ad click approaches on a real-world data set comprising hundreds of thousands of online sessions collected at Yahoo servers, with the empirical results strongly indicating benefits of the proposed methodology.

II. Background

A. Ad targeting of online users

A central problem in the online targeting industry is a task of predicting whether or not an online user would click or convert on a displayed ad. Despite this simple definition, the task is far from being naïve, having many challenging issues related to large scale, in addition to extreme diversity and complexity of the problem at hand [5]. Nevertheless, it has been shown that computational targeting, and in particular behavioral targeting, can result in statistically significant improvements in CTR [6], which directly translates into increased revenue for both the advertisers and the publishers. For these reasons, the task has garnered significant attention from the research community, as witnessed by a large number of recently proposed methods tackling this problem [9], [10], [11].

In most previously proposed approaches, e.g., as in [4], the authors use a common feature vector representation for both users and ads, derived from the historical browsing behavior or the immediate context, such as the words of the current web page or the issued query to a search engine for the users, and the text description for the ads. Then, a classifier is trained to estimate the probability of a user clicking on a particular ad, output of which is used to match users to ads that will be shown and that are most likely to be clicked on. Furthermore, more detailed user-specific features, such as geolocation or demographic information, have been shown to lead to improved performance [10].

However, aggregation of historical user-specific information at low resolutions [11] may be suboptimal, as short-term, highly granular feature representations could lead to significant performance improvements in the targeting tasks [6]. Further, modeling only the immediate context to estimate click propensity [9], while not explicitly modeling a sequence of actions that led to a desired outcome for the publisher, may also lead to unsatisfactory results. This is due to a fact that, quite intuitively, user's browsing path carries a strong signal related to user's intent and their future behavior [12]. In this work we consider an approach that accounts for this issues, and uses a short-term, user-generated path of actions to detect likely converting users with high accuracy.

B. Sequence modeling approaches

In order to make the sequence modeling problem tractable and the resulting models easily interpretable, a modeler usually introduces an assumption that the current state of the sequence depend only on the previous states, known as the Markov assumption. One approach to sequence modeling involves directly modeling the short-term transitions between observations using Markov chains [13]. However, these methods may be too simplistic for many real-world applications. To address these issues, one approach is to introduce hidden nodes to the model, where we assume that there exists a latent layer of nodes which affects the observation sequence. Hidden Markov Model (HMM) [14] is a particularly popular framework due to its efficiency as well as effectiveness. The HMMs are directly applicable to the task of modeling of online behavior, as can be seen from [15] where the authors show that low-order Markov models perform well when modeling actions of online users.

In addition to generative sequence modeling, there exist many popular and effective discriminative models, such as Conditional Random Fields (CRFs) [16] and their variants [7], [17], [18]. In the context of modeling of online behavior, there have been earlier attempts at employing CRFs to user understanding [17], [19]. However, unlike these approaches which aim at labeling each action within a session sequence with a label, in this paper we consider the problem of determining if a user is likely to click on an ad at a session-level, while still tracking the propensity of a click at the action-level. To this end, we propose to use HCRF [7], as detailed in the following section. Moreover, as the space of user actions is very large and thus prohibitive of the direct application of the considered sequence models, we address the sparsity issue by proposing to use a framework for distributed modeling of user actions. In particular, we extend the undirected architecture from [8] to model time dependency of user actions, mapping the actions to a compact lower-dimensional space, which resulted in a significant performance gains over the competing algorithms.

III. Distributed modeling of user actions

For a user u, we define a session $\mathbf{s}$ to be an uninterrupted sequence of actions $\mathbf{s} = [x_1, x_2, \dots x_T] \in \mathcal{S}$, where an action x_t represents online action at timestamp t (e.g., page visited, search performed), and $\mathcal{S}$ is the set of all sessions. We consider the sequence $\mathbf{s}$ to be uninterrupted if the duration between any two consecutive actions is less than a predefined threshold; in this work we used a threshold of 30 minutes [20]. To avoid sparsity issues due to a large cardinality of the action set $\mathcal{X}$, we represent each action x_t as a low-dimensional vector $\Phi(x_t) \in \mathbb{R}^D$, discussed in more detail in Section III-A.

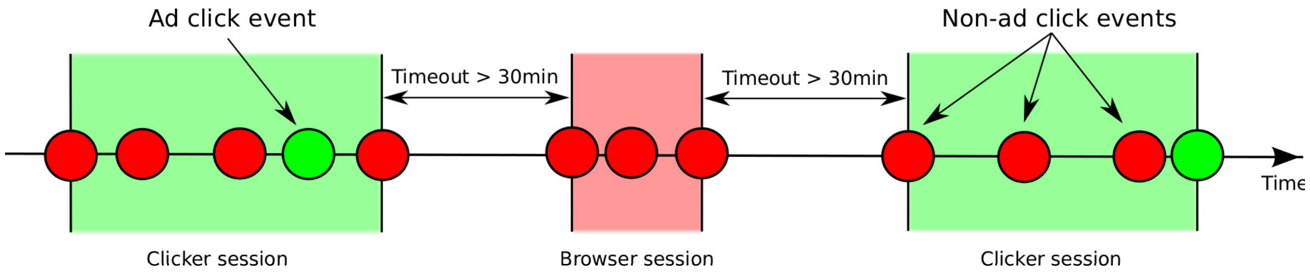


Fig. 1. Illustration of sessions: inactivity of more than half an hour defines boundary between sessions, and the entire session is labeled *clicker* if any of the user actions within that session is either an ad click or a purchase

Let $\mathcal{Y}$ be a set of possible user intent labels for a session. As we are interested in modeling a short-term sequence of user actions with respect to receptiveness to advertising, we define $\mathcal{Y}$ as a binary set $\mathcal{Y} = \{clicker, browser\}$, where $clicker$ represents a user session for which user receptiveness to advertising is high, and, conversely, $browser$ denotes a session for which user receptiveness is low. As a proxy of user intent and receptiveness to advertising within a session $\mathbf{s}$, we use the observed click information within the session. More formally, given a session $\mathbf{s}$, if any action within session $\mathbf{s}$ is an ad click or a purchase, we label the entire session as $clicker$, otherwise as $browser$. As stated earlier, our setup is such that the entire session gets assigned a single label, which is different from sequence modeling approaches discussed in Section II-B that consider different label for each action within a session. Our setup corresponds to real-world applications more realistically, since it is most often impractical or even impossible to acquire labels for every single action of a user. This is illustrated in Figure 1, where ad click and purchase actions are shown in green and other actions in red. If any of actions within a session is colored green then the entire session is labeled as $clicker$, otherwise as $browser$.

At the training time, given a training set $\mathcal{D}$ of N labeled sessions from different users $\mathcal{D} = \{(\mathbf{s}_i, y_i), i = 1 \dots N\}$ with $y_i \in \mathcal{Y}$, a learning objective is to find a mapping function $f : \mathcal{S} \rightarrow \mathcal{Y}$. At the prediction time, we use learned mapping $f(\cdot)$ to predict whether a new user session $\mathbf{s}$ is in a $clicker$ or a $browser$ state. In the remainder of this section we present the two main components of our proposed approach, the directed skip-gram model used to learn action representation $\Phi(\cdot)$, and the HCRF model used to estimate the prediction function $f(\cdot)$.

A. User action representation

Let us assume that the original dimensionality of the action space is D_h, and the desired low-dimensional space is D-dimensional, where typically $D_h \gg D$. Then, we can use recently proposed skip-gram model [8] to find a mapping from the input space to a D-dimensional embedding space. It was shown that this method can be successfully applied to find word representation in low-dimensional space such that semantically similar words appear as neighbors with respect to the cosine distance [8]. Motivated by these results, we propose a modification of the skip-gram suitable for finding distributed representation of user actions in the ad targeting tasks. Note that, as the skip-gram was originally developed for Natural Language Processing (NLP) tasks, in the following section we will use terms 'word' and 'action' interchangeably, keeping in mind that the approach is not limited to the NLP domain.

Directed skip-gram for temporal sequences The skip-gram model formulates representation learning as an unsupervised learning problem, and learns word embeddings such that in the embedding space the words that co-occur or occur in similar contexts are nearby. In NLP tasks it makes intuitive sense to exploit both the words before and after the central word when learning the embedding. However, in ad targeting the notion of time is of critical importance. For example, finding correlations between an ad click and search queries that occur after it is not as important as finding the correlations between an ad click and queries that precede it, since in the first case we have an opportunity to use the inferred correlations to steer the users towards a favorable action (e.g., ad click).

Following this reasoning, we propose a modification to the skip-gram model that considers past and future actions differently. In particular, given a sequence of user actions $\mathbf{s}$ and some central word x_t belonging to this sequence, the directed skip-gram model considers only l words that precede x_t. This yields the following optimization problem,

$$\text{maximize} \quad \frac{1}{T} \sum_{t=1}^{T} \sum_{1 \leq i \leq l} \log \mathbb{P}(x_{t-i} | x_t), \tag{1}$$

solved using stochastic gradient descent [8]. The directed skip-gram maximizes log-probabilities of word x_t given its preceding words, thus forcing the model to focus on predicting subsequent words instead of both past and future contexts.

B. Hidden CRF with distributed action embeddings

Given a user session $\mathbf{s}$ of lenght T, the HCRF models a conditional distribution of the session label y. However, in contrast to a standard CRF which assumes that each action is associated with its own label, HCRF [7] assigns a single label y to the entire session sequence, and introduces T latent variables $\mathbf{h} = [h_1, h_2, \dots, h_T]$ that are not observed in the training data, where each h_t corresponds to one user action x_t. Each h_t takes values from $\mathcal{H}$, a user-specified finite set of possible hidden states in the model. Intuitively, each latent variable h_t corresponds to a labeling of action x_t with one member of set $\mathcal{H}$, and can be used to discover dynamic hidden user intent within the session. Given these definitions of hidden states, HCRF models the following conditional distribution,

$$\mathbb{P}(y, \mathbf{h}|\mathbf{s}, \mathbf{w}) = \frac{1}{Z(\mathbf{h}, \mathbf{s}, \mathbf{w})} \exp\big(\Psi(y, \mathbf{h}, \mathbf{s}, \mathbf{w})\big), \qquad (2)$$

where $\mathbf{w}$ represents parameters of the HCRF model, $\Psi(\cdot)$ is the so-called potential function that defines relationships between the actions $\mathbf{s}$, latent variables $\mathbf{h}$, and the label y, and $Z(\cdot)$ is a normalization function, computed as

$$Z(\mathbf{s}, \mathbf{w}) = \sum_{y', \mathbf{h}} \exp\big(\Psi(y', \mathbf{h}, \mathbf{s}, \mathbf{w})\big). \qquad (3)$$

Then, it follows that

$$\mathbb{P}(y|\mathbf{s}, \mathbf{w}) = \sum_{\mathbf{h}} \mathbb{P}(y, \mathbf{h}|\mathbf{s}, \mathbf{w}) = \frac{\sum_{\mathbf{h}} \exp\big(\Psi(y, \mathbf{h}, \mathbf{s}, \mathbf{w})\big)}{\sum_{\mathbf{y}', \mathbf{h}} \exp\big(\Psi(y', \mathbf{h}, \mathbf{s}, \mathbf{w})\big)}. \qquad (4)$$

The HCRF potential is defined as a linear combination of feature functions, similarly to the standard CRFs,

$$\begin{aligned}\Psi(\mathbf{h}, y, \mathbf{s}, \mathbf{w}) = & \sum_{t=1}^{T} \sum_{k=1}^{|\mathcal{H}|} \sum_{d=1}^{D} w_{kd}^{(1)} f_{kd}^{(1)}(x_t, h_t) + \\ & \sum_{t=1}^{T} \sum_{k=1}^{|\mathcal{H}|} \sum_{r=1}^{|\mathcal{Y}|} w_{kr}^{(2)} f_{kr}^{(2)}(y, h_t) + \\ & \sum_{t=2}^{T} \sum_{k_1, k_2=1}^{|\mathcal{H}|} \sum_{r=1}^{|\mathcal{Y}|} w_{k_1 k_2 r}^{(3)} f_{k_1 k_2 r}^{(3)}(y, h_{t-1}, h_t).\end{aligned} \qquad (5)$$

The first feature function models compatibility of action representations and hidden action labels through

$$f_{kd}^{(1)}(x_t, h_t) = \Phi_d(x_t) \cdot I(h_t = k), \qquad (6)$$

where $\Phi_d(x_t) \in \mathbb{R}$ is the d^{th} component of a D-dimensional action representation $\Phi(x)$, and $I(\cdot)$ is an indicator function that equals 1 if the argument is true and 0 otherwise. The compatibility of each component of action representation with each hidden state will be determined by the corresponding parameter $w_{kd}^{(1)}$ in (5). There are $D \cdot |\mathcal{H}|$ such parameters. Note that in the naïve one-hot representation approach the parameter space would grow prohibitively large, resulting in serious sparsity problems. By using the proposed distributed embedding of user actions such issues are mitigated [21].

The second type of feature functions model compatibility between hidden state h_t and session-level label y as follows,

$$f_{kr}^{(2)}(y, h_t) = I\big((h_t = k) \wedge (y = r)\big), \qquad (7)$$

with $|\mathcal{Y}| \cdot |\mathcal{H}|$ corresponding parameters $w_{kr}^{(2)}$ from (5).

Finally, the last type of feature function models the compatibility between neighboring hidden states and the label,

$$f_{k_1 k_2 r}^{(3)}(y, h_{t-1}, h_t) = I\big((h_{t-1} = k_1) \wedge (h_t = k_2) \wedge (y = r)\big), \qquad (8)$$

with $|\mathcal{Y}| \cdot |\mathcal{H}|^2$ corresponding parameters $w_{k_1 k_2 r}^{(3)}$ from (5).

Training of HCRF comprises maximizing log-likelihood over the training data given by the following equation,

$$\log \mathbb{P}(\mathbf{y}|\mathbf{s}, \mathbf{w}) = \sum_{i=1}^{N} \log \mathbb{P}(y_i|\mathbf{s}_i, \mathbf{w}) - \lambda \mathbf{w}^{\mathrm{T}} \mathbf{w}. \qquad (9)$$

Due to a linear-chain structure of the model, exact methods exist for parameter estimation. We follow [7] and use a conjugate-gradient optimization method to find parameters $\mathbf{w}$ that maximize the log-likelihood from (9), where we set $\lambda = 1$. Lastly, in the inference phase we use the following expression,

$$\hat{y} = \operatorname*{argmax}_{y} \Big(\sum_{\mathbf{h}} \mathbb{P}(y, \mathbf{h}|\mathbf{s}, \mathbf{w}) \Big). \qquad (10)$$

IV. Experiments

The considered data set was generated using anonymized information about users' online actions collected at Yahoo servers. The actions are temporal sequences of raw events extracted from server logs and are represented as tuples $(u_i, a_i, t_i), i = 1, \ldots, N_{logs}$, where u_i is an ID of a user that generated the i^{th} tuple, a_i is an activity log, t_i is a timestamp, and N_{logs} is a total number of tuples. We collected logs belonging to one of the five types over a one-month period:

- page views ("pv") - website pages that the user visited;
- search queries ("sq") - user-generated search queries;
- sponsored link clicks ("slc") - user clicks on search-advertising links that appear next to search links;
- ad clicks ("adc") - display ads that the user clicked on;
- receipts ("prch") - e-mail purchase receipts.

The logs belonging to the "pv" and "sq" groups were used to generate user actions x. In particular, for "pv" we used entities found on a webpage as words[1], while for "sq" we used tokens in a query as words, which were chronologically ordered and sent as an input to the directed skip-gram model. We used publicly available code for the skip-gram[2] which we modified to implement the directed version, where we set the $D = 100$. Furthermore, we used "slc" and "adc" actions to find the ad click labels for sessions, and "prch" actions to find conversion labels, and did not use these actions as inputs to the directed skip-gram. Note that the two types of labels were used separately for two sets of experiments, ad click and purchase prediction. We removed sessions shorter than 5 actions, and in the experiments used a balanced data set with 478,861 online sessions generated by 215,417 users.

For illustration, in Figure 2 we give examples of *clicker* and *browser* sessions. We can see that the *browser* session was not as focused on the subject matter as the *clicker* session (i.e., jumping between "baby" and "iphone" queries), and the user was only interested in satisfying their immediate information need (i.e., "how to"). On the other hand, the *clicker* session started with general terms which got refined with the follow-up queries (i.e., from "halloween" to "star trek officers uniform"), eventually resulting in an ad click. This is a pattern which we aim to capture with the proposed sequence modeling approach.

[1] nlp.stanford.edu/software/CRF-NER.shtml, accessed October 2014

[2] code.google.com/p/word2vec/, accessed October 2014

Queries in *browser* session

1) how to feed a baby
2) how to keep a baby quiet
3) how to update ur iphone 5
4) how to update your iphone 5
5) how to update ur iphone 5

Queries in *clicker* session

1) halloween city stores
2) female star trek characters
3) female star trek officers
4) female star trek officers
5) star trek officers uniform

Fig. 2. Examples of user sessions (actions ordered chronologically)

A. Competing prediction models

Logistic regression (LR) and Support vector machines (SVM). We first considered linear LR and non-linear SVM classifiers, which both assume that actions within a user session are independent. Since LR and SVM can not encode the dynamics between user actions in the session, the data set was processed in such a way that all feature representations within a session were exponentially time-decayed and aggregated into one feature representation $\Phi(\mathbf{s})$ for session $\mathbf{s}$,

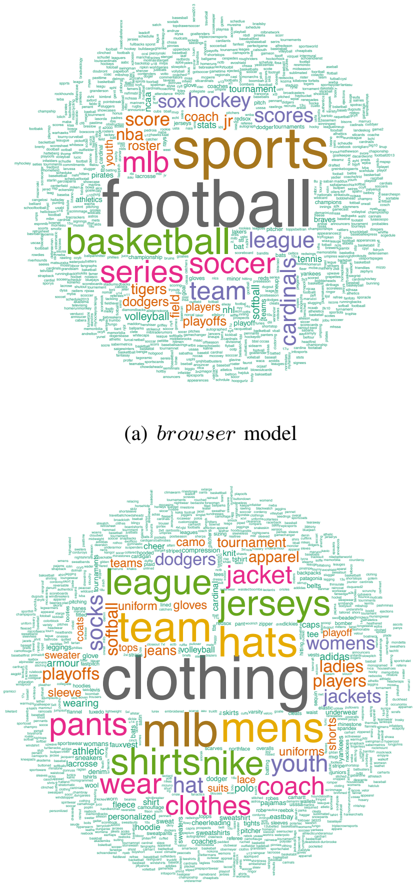
$$\Phi(\mathbf{s}) = \sum_{t=1}^{T} \Phi(x_t)\alpha^{t_T - t_t},$$

where t_t is the timestamp of the t^{th} action, and α is a decay factor with $0 < \alpha \leq 1$ (we set this parameter to 0.99). In the experiments we used Vowpal Wabbit toolbox[3] for logistic regression and LibSVM toolbox [22] for SVM. We used a Gaussian kernel in SVM, and randomly split the data set using 70% of examples for training and the remainder for testing.

Hidden Markov Model (HMM). We used publicly available implementation[4], and trained one HMM model per each class: the *clicker* model trained on all sessions labeled as *clicker*, and the *browser* model on all sessions labeled as *browser*. Both models had 2 hidden states and the observation model assumed multivariate Gaussian distribution. During testing phase, a test session was passed through the both HMM models. Then, a label associated with HMM that resulted in a higher likelihood was selected as the predicted label [23].

Conditional Random Field (CRF). The model learns to predict labels for each action within a training session. In order to be able to train CRF, we assigned an overall session label to all actions within that session. During testing, we run the Viterbi algorithm on test sessions to find the most likely label sequence, assigning the most frequent label to a session [23].

Hidden Conditional Random Fields (HCRF). We used the HCRF model described in Section III-B, setting the cardinality of the latent set to $|\mathcal{H}| = 2$. During testing, a session was labeled with a label that maximized equation (10). For both CRF and HCRF we used a publicly available code[5].



(a) *browser* model

(b) *clicker* model

Fig. 3. Word cloud of nearest neighbors for word ”baseball” obtained by the directed skip-gram trained on: a) *browser* sessions; b) *clicker* sessions

As our main goal is to detect *clicker* sessions, in order to evaluate effectiveness of the considered methods we used precision (fraction of true *clicker* sessions among the predicted ones), recall (fraction of correctly retrieved *clicker* sessions), and F1-score measures. Ideal method should have high values for both precision and recall, although in targeted advertising only CTR (i.e., precision) is commonly measured and reported.

B. Results

We first demonstrate that feature representation of user actions obtained by the directed skip-gram model is able to capture subtle differences in user behavioral patterns in the *clicker* and *browser* modes. To illustrate this, we used two directed skip-gram models: one trained on *clicker* sessions, and the other trained on *browser* sessions in the same time period. In Figures 3(a) and 3(b) we show word clouds of the nearest neighbor words in the inferred lower-dimensional embedding space, with respect to cosine distance, of term ”baseball” for the *browser* and *clicker* skip-gram models, respectively (note that, as we used only ”pv” and ”sq” user actions, we were able to plot word clouds as inferred by our models). We can see that the nearest neighbors for ”baseball” in the *browser* model were all related to general sports terms,

[3] github.com/JohnLangford/vowpal_wabbit, accessed October 2014

[4] www.run.montefiore.ulg.ac.be/~francois/software/jahmm/, acc. Oct. 2014

[5] sourceforge.net/projects/hcrf/, accessed October 2014

TABLE I

Relative improvement (in %) with respect to logistic regression (averaged over 5-fold cross-validation runs)

		SVM	**HMM**	**CRF**	**HCRF**$_{undir}$	**HCRF**$_{dir}$
Click pred.	Prec.	7.82	1.03	3.89	9.06	12.9
	Rec.	8.84	1.10	3.59	9.85	11.1
	F1	8.29	1.06	3.74	9.39	12.1
Purch. pred.	Prec.	1.03	0.17	0.86	2.07	2.58
	Rec.	2.04	0.41	0.20	2.65	3.47
	F1	1.57	0.30	0.50	2.38	3.06

such as ”football”, ”sports”, ”basketball”. On the other hand, we see that most of the nearest neighbors in the $clicker$ model were terms that can be associated with purchasing behavior (e.g., ”clothing”, ”hats”, ”pants”). These results provide a strong evidence that the skip-gram model can find useful and intuitive embeddings, and that the sequential patterns of user actions differ significantly between the two modes, confirming findings from Figure 2. In the remaining experiments we used representation of user actions obtained by training a single skip-gram model using the entire training data set.

In the next set of experiments we evaluated predictive power of the models. We conducted experiments using 5-fold cross-validation on two versions of the data set: 1) ad click prediction data, where the labels were obtained using user ad clicks; and 2) purchase prediction data, where the labels were obtained using e-mail receipts. In order to protect business-sensitive information, we only give relative performance improvement over the logistic regression, and report the precision, recall, and F1-score in Table I. By considering the last two columns, we can see that the proposed directed skip-gram outperformed undirected approach (although we report this result only for HCRF, similar advantage of directed over undirected skip-gram was found for the competing methods as well). This is an expected result, due to the fact that there exists a strong temporal causality in user online behavior, which can not be fully captured by the undirected skip-gram. Consequently, we only used representations obtained by the directed skip-gram to report the results of the baseline methods.

Further, we can see that a simple HMM model obtained only a small improvement over logistic regression. More expressive CRF resulted in further improvement of performance, however this model is not suitable for sequence-level labeling, and may have suffered from the labeling of user actions during training using a session-level information. Interestingly, even though it does not model sequential data explicitly, SVM obtained very competitive performance, even outperforming HMM and CRF. Lastly, we see that HCRF obtained the best results.

By comparing the reported results for click and conversion prediction given in Table I, we can see that the problem of conversion prediction posed much bigger challenge. Although HCRF still outperformed the competition and obtained nearly 3% improvement over the logistic regression, it is an interesting research avenue to develop methods that would obtain superior performance in this case as well, as user conversions are what the advertisers are ultimately interested in.

V. Conclusion

We considered the problem of estimating user's propensity to click on an ad or make a purchase, a critical problem in ad targeting. We predicted whether a user in a particular session is a $clicker$ or just a $browser$, indicating higher or lower responsiveness to the advertising campaign, respectively. The results showed that the proposed directed skip-gram architecture found useful representations of user actions, and that HCRF significantly outperformed the baseline approaches.

References

[1] D. S. Evans, “The online advertising industry: Economics, evolution, and privacy,” *The journal of economic perspectives*, pp. 37–60, 2009.

[2] D. C. Fain and J. O. Pedersen, “Sponsored search: A brief history,” *Bulletin of the American Society for Information Science and Technology*, vol. 32, no. 2, pp. 12–13, 2006.

[3] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, “Accurately interpreting clickthrough data as implicit feedback,” in *SIGIR*, 2005, pp. 154–161.

[4] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola, “Scalable distributed inference of dynamic user interests for behavioral targeting,” in *KDD*, 2011, pp. 114–122.

[5] H. B. McMahan, G. Holt *et al.*, “Ad click prediction: A view from the trenches,” in *KDD*, 2013, pp. 1222–1230.

[6] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, “How much can behavioral targeting help online advertising?” in *WWW*, 2009.

[7] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, and M. Csail, “Hidden-state conditional random fields,” *IEEE TPAMI*, vol. 29, no. 10, pp. 1848–1852, 2007.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013, pp. 3111–3119.

[9] Y. Chen and T. W. Yan, “Position-normalized click prediction in search advertising,” in *KDD*, 2012, pp. 795–803.

[10] H. Cheng and E. Cantú-Paz, “Personalized click prediction in sponsored search,” in *WSDM*, 2010, pp. 351–360.

[11] T. Wang, J. Bian, S. Liu, Y. Zhang, and T.-Y. Liu, “Psychological advertising: Exploring user psychology for click prediction in sponsored search,” in *KDD*, 2013, pp. 563–571.

[12] B. D. Davison and H. Hirsh, “Predicting sequences of user actions,” in *Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis*, 1998, pp. 5–12.

[13] J. Borges and M. Levene, “Evaluating variable-length Markov chain models for analysis of user web navigation sessions,” *IEEE TKDE*, vol. 19, no. 4, pp. 441–452, 2007.

[14] L. Rabiner and B.-H. Juang, “An introduction to hidden Markov models,” *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[15] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlós, “Are web users really Markovian?” in *WWW*, 2012, pp. 609–618.

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

[17] Y. Shen, J. Yan, S. Yan, L. Ji, N. Liu, and Z. Chen, “Sparse hidden-dynamics conditional random fields for user intent understanding,” in *WWW*, 2011, pp. 7–16.

[18] V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Continuous conditional random fields for regression in remote sensing,” in *ECAI*, 2010.

[19] Q. Guo, E. Agichtein, C. L. Clarke, and A. Ashkan, “In the mood to click? Towards inferring receptiveness to search advertising,” in *IEEE/WIC/ACM WI-IAT*, vol. 1, 2009, pp. 319–324.

[20] P. L. Pirolli and J. E. Pitkow, “Distributions of surfers' paths through the World Wide Web: Empirical characterizations,” *World Wide Web*, vol. 2, no. 1-2, pp. 29–45, 1999.

[21] Y. Bengio and Y. Lecun, *Scaling learning algorithms towards AI.* MIT Press, 2007.

[22] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for Support Vector Machines,” *ACM TIST*, vol. 2, no. 3, p. 27, 2011.

[23] S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell, “Hidden conditional random fields for gesture recognition,” in *IEEE CVPR*, vol. 2, 2006, pp. 1521–1527.