Big Data
Volume 00, Number 00, 2022

DOI: 10.1089/big.2021.0320

Predicting Actions of Users Using Heterogeneous Online Signals

Djordje Gligorijevic,*,† Jelena Gligorijevic, and Aaron Flores

Abstract

Advertising platforms have a growing need for improving prediction quality, as missing out on ad opportunities can have a negative effect on their performance. To that end, prediction tasks such as conversion prediction need to be continuously advanced through the inclusion of data from new sources or through algorithmic development that tackles existing challenges. The introduction of different data sources naturally brings unwanted noise, whereas underexplored areas still exist in modeling approaches, such as temporal information of events in sequences. In this study, we propose extensions for modeling online user activity trails that address two very important aspects of activities—time and noise, through dedicated layers that can be used in existing deep sequence-learning approaches. Our proposed method exhibited area under the receiver operating characteristic curve improvement of up to 3% and 1.75% compared with production and best baseline approaches, respectively, across two major advertiser data sets and several predictive tasks.

Keywords: computational advertising; deep learning; discretized representations; time-aware modeling; user activity trails

Introduction

The challenge of online display advertising (DA) lies in displaying the most relevant advertisements (ads) to the right users anywhere online in a timely manner. In this industry, continuous improvement of user modeling for maintaining competitive key performance metrics has always been the key factor of success for advertising platforms. Ads platforms strive toward learning as much as possible about users' interests from their online behavior to recognize the right opportunities to display attractive ads to users. This is a key component in making the entire DA ecosystem efficient[1] and allowing it to grow to hundreds of billions of dollars worldwide.*

Modeling interests of a user heavily depends on the data available about them, and ads platforms will resort to collecting activities on a user from a wide spectrum of sources such as advertisers' websites traffic, won online auctions, third-party data, and from owned-and-operated (O&O) properties. The activities can span very different types of data such as search history, news articles read, e-mail and mobile stores purchase receipts, various mobile events, and geo locations. Furthermore, interests and preferences of users are learnt by aligning the compiled users' online footprint with advertisers' products, finally identifying users that could become their business in the near future.

An example of one such online footprint is provided in Figure 1 where we observe multiple interactions a user had with different properties online, such as mobile and desktop search, reading news, and interacting with advertisers' websites before a conversion action, or purchase in the example. Actions are the most general term for activities of interest highlighted by advertisers and include conversions (such as online order or conversion), but also other events such as user visiting the

*https://www.iab.com/wp-content/uploads/2020/05/FY19-IAB-Internet-Ad-Revenue-Report_Final.pdf (last accessed October 2021).

Yahoo! Research, Sunnyvale, California, USA.
†Current affiliation: *eBay, San Jose, California, USA.*

*Address correspondence to: *Djordje Gligorijevic, eBay, 2145 Hamilton Avenue, San Jose, California, USA, E-mail:* gligorijevic@temple.edu

FIG. 1. User activity trail sequence with different types of activities ordered by the time they occurred and ending with the action of an advertiser's interest.

advertiser's website for the first time (retargeting event), or inquiries or interactions with advertisers' websites such as clicks or reads of advertisers' content. The methodology proposed in this article is suitable for any actions users may take that advertisers highlight as important.

Learning user's interests from data collected across a variety of sources is a nontrivial task. First of all, the number of unique activities collected may range from hundreds of thousands to billions, thus making feature selection and engineering very tedious tasks. Second, coming from different data sources and with the magnitude and diversity of potential actions, not all the signals will have the same predictive strength across many different prediction tasks. This is what we refer to as the heterogeneous property of activity and it is an important aspect of our study. Third, due to a variety of different predictive tasks (per advertiser, per ad campaign, etc.) it is difficult to highlight noninformative signals.

Furthermore, these trails of user's activities provide insight into the sequence of actions carrying more information than sequence-oblivious features. Moreover, these trails are not just uniform sequences of tokens, user actions always come with assigned timestamps, which also carry a significant amount of additional information in terms of how close the subsequent activities were or how much time passed between activity and event of interest (i.e., conversion).

Despite these challenges, large sets of well-designed features are traditionally created through manual curation and feature engineering in the industry, thus partially overcoming the challenges. Using these features machine learning models have been successfully applied for different predictive tasks (i.e., click or conversion predictions). Despite the existing success, designing and selecting appropriate features for different tasks remain a very challenging problem.[2]

Significant advances with the aforementioned challenges have been attained more recently, with the efforts in developing deep representations of activities to help automatically learn their features.[3,4] Even though these methods model sequences of activities and apply different strategies to tackle or filter noisy data such as various attention mechanisms, there are several challenges remaining largely untackled in the practice. We highlight two emerging ones in this study.

The first one is the temporal information that comes with sequences where activities do not occur uniformly in time. The second is inattentiveness to the inherent noise in the data collected from different sources, which traditional machine learning solutions normally assign uniform focus to all tokens in a sequence that are not filtered out by any preprocessing of feature selection techniques. Even when existing approaches do try to model noise through different attention mechanisms,[5] they usually are not able to completely remove the influence of noisy activities, but merely lower their effect in the overall prediction. In the context of this study, we call such attentions soft attentions.

To address these two issues, we (1) describe existing work in including temporal information and exploit a recent solution to capture the temporal aspect of sequences, and (2) propose hard attention mechanism to address the ever-present issue of having random noise in available signals to further boost algorithmic performance.

Key contributions of this study are summarized as follows:

- We model conversion prediction task based on time-ordered sequences of users' activities collected from multiple data sources.
- We exploit a time-aware mechanism to capture the temporal aspect of activities. The approach accumulates up to 2.4% against production and 1% against the best baseline area under the receiver operating characteristic curve (AUC) lifts.
- We propose a hard attention mechanism to help with untreated noise contained in heterogeneous

data. The proposed approach accumulates up to 1.8% and 0.5% AUC lifts, whereas the combined approach with time accumulates up to 3% and 1.75% AUC lifts against production and the best baseline, respectively.

The rest of the article is organized as follows. In Background and Related Work section we describe previous approaches to model conversions in DA as well as recent advances from which we motivate our modeling choices. The proposed methodology is discussed in Methodology section. Data sets used to evaluate the proposed method, baselines, and evaluation metrics are described in Data Description section. Experimental results are discussed in Experimental Evaluation section and finally, study is concluded in Conclusions section.

Background and Related Work

Importance of predicting future actions of users, such as conversions, is first given in the context of online ads platforms ecosystem. In addition, relevant prior studies on conversion and click prediction tasks are discussed stressing important aspects that motivated this study. Finally, approaches tackling two important and underexplored aspects of online user modeling discussed in the introduction, sequence temporal, and signal noise modeling are discussed in detail as they are major building blocks of the proposed approach.

Importance of action prediction to online advertising systems

Running online campaigns on behalf of advertisers is the main task of major ads platforms for DA. These campaigns are designed to target certain activities such as clicks or conversions (i.e., online purchase, booking, or service subscription). Ads platforms participate in online auctions where they bid for opportunities to show ads to users who are a potential business on behalf of advertisers, thus achieving satisfactory key performance indicators.

Once decided which ad should be shown to a user for a given ad opportunity, platforms need to specify their bid and participate in the online auction. In a simplified manner, a bid is in most cases decided using the following or similar formulation[6]:

$$bid_i = f(\alpha * pCVR_i * impression\_value_i), \quad (1)$$

where α is a product of several control parameters such as pacing, Cost-Per-Mille, Cost-per-Click, Cost-Per-Action, and other controllers and multipliers, $pCVR_i$ is predicted conversion rate (example of action) for the given opportunity i and $impression\_value_i$ is the dollar value of each impression set manually by the advertiser at line creation time. Function f represents the dominant bidding strategy.

Putting it all together, deciding the value of the bid to submit in online ad auction has three key aspects. Thus, estimating action, that is, conversion, probability plays an integral role that drives performance and allows an ad system to display ads to relevant users.

Modeling users' conversion prediction

Linear models, such as logistic regression,[7] or nonlinear models, such as random forests (RF), have been successfully utilized for tackling conversion probability estimation in large advertising platforms. These approaches run on powerful syntactic or semantic handcrafted features.[8]

However, the downside is that these approaches inherently rely on manual design and selection of features, which requires a substantial investment of human time and effort. The usefulness of such handcrafted features is largely dependent on the domain knowledge of human experts curating the features. With the rise of DA, there has been an expansion of these domains ranging across retail, automotive, travel, communication, and so on, thus expanding the necessary knowledge of handcrafting features to ensure high performance. Moreover, predictive tasks are nonlinear in the feature space, but considering feature interactions (e.g., cross-features) quickly becomes prohibitively expensive due to a combinatorial explosion.[9]

To address the aforementioned shortcomings, several research directions with representation learning capabilities have been proposed, notably factorization machines[10] for conversion prediction or deep residual networks[11] and Siamese networks[12] for click prediction that tackle problem of learning nonlinear interactions of raw features. Deep neural networks were also proposed to learn representations from traditional features.[11,13,14]

Finally, as discussed, the online data about users are naturally collected as sequences of activities thus, models that capture information from the sequence, most notably recurrent neural network (RNN)-based deep architectures, have been proposed.[3,14–16] A common theme for such approaches is that they perform significantly better than their nonsequential counterparts unraveling the fact that the sequence of information contains additional information that was previously

largely underexploited. Moreover, Arava et al.[16] and Zhou et al.[4] have used sequences of activities from multiple data sources, whereas Arava et al.[16] have additionally tackled the problem of conversion attribution task for which they utilized temporal information of activities.

Exploring temporal information in activity sequences

Modeling timestamp information in sequences has been largely unexplored in the literature for a long time; however, more recently several formulations have been proposed. The authors of Pei and Tax[17] and Beutel et al.[18] propose several ways of generating time features to be added to existing feature set. Examples of time features generated from activity timestamp t (with additional hyperparameter γ) include linear features: $v_t = \frac{t}{T}$, delta features[18]: $v_t = t - T$, tanh features: $v_t = \tanh(\gamma + \frac{t}{T}) + 1 - \tanh(\gamma)$ and exp features: $v_t = \exp(\gamma \frac{t-T}{T})$. However, these are the cases of strict time decay effect where only activities that happened close to prediction time may have higher values.

Categorizing time information has also been proposed, such as categories of time differences between activities (i.e., short or long)[19] or hour of activity.[20]

Extending long short-term memory cells by adding additional gate that takes into account the time passed between subsequent events also showed promising results,[21] however, without taking into account event-specific temporal patterns. Another way of using time information was proposed in Arava et al.[16] with attention regularization mechanism that penalizes embeddings from being similar for two activities that occur at a larger temporal distance. Finally, Rajkomar et al.[22] propose generating time features, similarly to Pei and Tax,[17] whereas the features are not appended to the existing feature set, but used for generating attentions of different activities through softmax layer.

These approaches, however, ignore activity-specific time aspects of temporal transition of an activity, whereas the majority exclusively model the time decay factor penalizing activities that happen earlier in the user trail that could have a long-term effect. With these limitations of existing approaches, modeling temporal information in activity trails would be incomplete.

Activity-specific temporal modeling based on state transition modeling of dynamic systems was proposed for a special case of conversion prediction task.[23] In this study, we describe and use the time-aware attention model to capture the complex information from timestamps of activities in users' trails.

Exploring stochastic hard attentions for modeling noisy data

In general, machine learning framework information of the input is treated uniformly without discrimination of different parts. This process is different from human reasoning that tends to selectively concentrate on a part of information and at the same time filter out a portion of perceivable information. To address this uniform focus across different inputs, many variants of attention mechanism were proposed[24–27] for both sequence and image modeling with improvements over their nonattention baselines.

However, all of the existing approaches, limited by the need of learning differentiable continuous representations, learn attentions in such a way that all fields of feature maps will be given nonzero values, thus they have no benefit of completely removing signals in the feature map that may be noise. Addressing this challenge a differentiation between existing attention mechanism (named soft attentions in the remainder of the text) and attentions based on discrete representations capable of modeling zero weights (named hard attentions) has been recently proposed.[27] This initial study successfully compared hard versus soft attention mechanism on the image captioning task. Moreover, discrete representations have shown benefits in modeling sequence data that are inherently discrete.[28,29]

Discrete representation, however, requires stochastic neurons that traditionally could not be trained through conventional backpropagation algorithm. Thus, policy gradient[30] algorithm, an unbiased gradient estimator, was employed for the task. However, this algorithm suffers from being very complicated to implement and often yields high gradient variance during training.[31]

More recently, techniques such as reparametrization trick[32] allowed for the development of novel discrete units that could be optimized through backpropagation technique removing much of the shortcomings existing approaches had, bringing to development of Gumbel-softmax[33,34] and semantic hashing[35] techniques. Building on the novel advances, hard attention was extended yielding better performance and more stable optimization on several computer vision tasks[36,37] and unsupervised sequence representation learning.[38]

Hard attention mechanism was successfully employed using Gumbel-softmax technique on several computer vision tasks[36,37] by forcing attention weights of the feature maps to be zero selecting only a few or only a single value from the entire feature map.[37] This approach

allowed to address the inherent noise in the data and force algorithm to summarize feature map information effectively into a scalar. Modeling hidden layers of neural networks to be discrete[38] in addition to hard attention modeling has shown benefits in allowing improved reasoning from the discrete hidden representations.[28] In this case, semantic hashing technique was shown to be superior.

Furthermore, Gumbel-softmax approach was successfully applied to model compression and sparsification tasks as a differentiable $\mathcal{L}_0$ regularization framework to improve model training and inference speed and generalization.[39]

As both Gumbel-softmax and semantic hashing techniques were shown to be successful on different tasks, we describe both techniques in detail and empirically evaluate their performance on the sequences of activities from a large universe of diverse activities for the activity or conversion prediction task.

Discretization through Gumbel-softmax. Gumbel-softmax is a categorical reparametrization technique for a smooth approximation to Bernoulli random variables that allows efficient estimation of discrete units during training of neural networks, which was shown to be efficient and has high performance.[33,34]

Gumbel-softmax works as follows. Given a vector h_i, samples g_i are drawn first from the Gumbel distribution: $g_i = -\log(-\log(u))$, where $u \sim \mathcal{U}(0,1)$ are uniform samples. Then, Gumbel-softmax samples are drown from the softmax as

$$y_i = \frac{\exp(\log(W_g h_i + b_g + g_i)/\tau)}{\sum_{j=1}^{k} \exp(\log(W_g h_j + b_g + g_j)/\tau)}, \quad (2)$$

W_g and b_g are learnable parameters, k indicates dimension of generated softmax vector, whereas hyperparameter τ is a temperature parameter whose value dictates the sparseness of the resulting Gumbel-softmax distribution vector. With low temperature τ resulting vector is close to 1-hot vector (i.e., with $\tau = 0$ original Bernoulli would be recovered, but differentiability would be lost), and with large values of τ resulting vector resembles uniform distribution.[33] As τ is sensitive parameter that can be crucial performance-wise, adaptive temperature strategy[31] is applied to learn it with the remainder of network's parameters. Specifically, the following mechanism is set to determine the temperature value:

$$\tau = \frac{1}{softplus(W_{temp} h_i + b_{temp}) + 1}, \quad (3)$$

with W_{temp} and b_{temp} being dedicated parameters for temperature helping in disentanglement of the network, whereas adding 1 can enable the temperature to fall in the score of 0 and 1.

Discretization through semantic hashing. Discretization of vectors is also possible using semantic hashing technique.[35] Similarly to Gumbel-softmax, semantic hashing allows avoid annealing of the noise and provides a stable discretization mechanism that does not require additional loss factors. In this technique, to discretize vector h_i, during training exclusively, Gaussian noise is first added $h_n = h_i + n_i$, where $n_i \sim \mathcal{N}(0,1)$, whereas the sum operation is element-wise. From h_n two vectors are computed: $h_n^{(1)} = \hat{\sigma}(h_n)$ and $h_n^{(2)} = (h_n < 0) \in [0,1]$, where $\hat{\sigma}$ is the hard-sigmoid function[38]:

$$\hat{\sigma} = max(0, min(1, \sigma(x)(\gamma - \beta) + \beta)), \quad (4)$$

with $\sigma(x)$ "stretched" to the (β, γ) interval, with $\beta < 0$ and $\gamma > 1$ (i.e., $\beta = -0.1$ and $\gamma = 1.1$).

Discretized value of vector h is annotated as b-dimensional vector h^b. h^b will have b bits interpreted as integers between 0 and 2^b. Thus, the value b could be chosen carefully with respect to the dimensionality of the data used, that is, if there are ~ 1000 unique features, b should be chosen to be 10 as $2^{10} - 1 = 1023$.

Methodology

Building on the related work with the goal of addressing existing challenges, we propose a novel deep architecture for conversion rate estimation that adopts three attention mechanisms: SofT, hARd, and Temporal, a framework that we refer to as the START model.

The START model (shown in Fig. 2) takes two sets of inputs: sequence of activities $\{e_i | i = 1 \ldots N\}$ and time differences of activities' timestamps and the time point of prediction. Activities from different data sources are embedded into joint space, whereas temporal distances are used for learning temporal attention. After combining the two, the new representation is passed to a recurrent block to capture the sequence of activities information. Soft attention and hard attention, learned through a discrete net, are used in parallel and their combination produces a vector representation of the input used to predict by a final classification block.

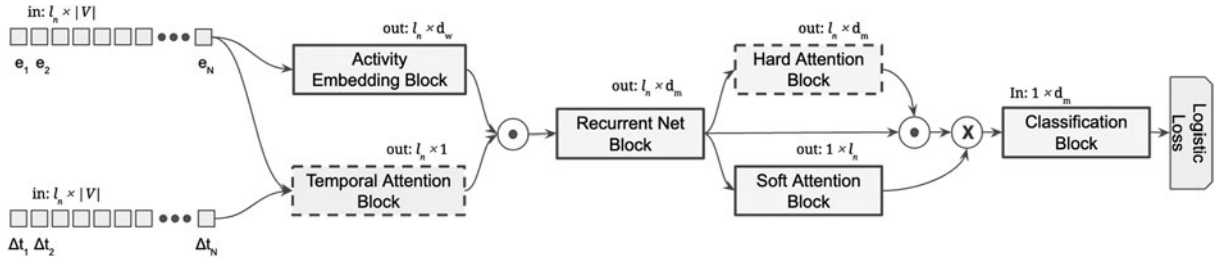

FIG. 2. Graphical representation of the proposed START model. Proposed blocks are highlighted in dashes and element-wise multiplication with feature maps from other blocks shows the ease of inclusion of such blocks in existing performing models such as attention RNN model shown through solid blocks. RNN, recurrent neural network.

Blocks of the proposed model

We now describe blocks of the START model in detail and provide hyperparameter information. All hyperparameters are chosen at a balance between computational and predictive performance, given available infrastructure and previously reported values.[12,23]

Activity embedding block. Activities $\{e_i | i = 1 \ldots N\}$ in the user's trail are embedded into common space vectors $h_{e_i} \in \mathbb{R}^{d_w = 200}$.

Temporal attention learning block. Motivated by the Euler's forward method[40] for modeling change of state in dynamic linear systems, we treat each activity as a system whose state changes through time, as done for temporal attention modeling in Gligorijevic et al.[23]

To model temporal information of the activity, forward method formulation is reused and its state value is squashed to a probability using sigmoid function to obtain activity-level contribution to the final task. Two single-dimensional parameters are associated to each activity e_i: $\mu_{e_i} \in \mathbb{R}^{d_t = 1}$ and $\theta_{e_i} \in \mathbb{R}^{d_t = 1}$. These parameters are designed to model the temporal increment Δ_t as time difference between current state i and the state of interest j (i.e., timestamp when prediction is used in ads system), capturing the important factor of timeliness of the prediction:

$$\Delta_t = \tau_{e_j} - \tau_{e_i}. \tag{5}$$

$$\delta(e_i, \Delta_t) = S(\theta_{e_i} - \mu_{e_i} \Delta_t), \tag{6}$$

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{7}$$

parameter θ_{e_i} measures initial (time-invariant) influence, whereas μ_{e_i} measures the change of the influence (time-variant) of the activity with the time difference. Activities whose influence does not change as we observe the activity through different points in the users trails will have small $|\mu_{e_i}|$, whereas the opposite means that position and time of the activity is very important for measuring its effect on conversion probability. As Δ_t is always positive and provided that θ_{e_i} does not change, larger positive values of μ_{e_i} would mean that temporal score is closer to 0, and larger negative values that is closer to 1. Similarly, large positive values of $|\theta_{e_i}|$ refer to stronger time-invariant impact of the activity.

In contrast to existing approaches to modeling temporal information in sequences described in Exploring Temporal Information in Activity Sequences section, our approach learns activity-specific temporal factors.[41] These factors are used for gating how much information passes from each activity embedding into the first nonlinear layer using the sigmoid activation. Furthermore, impact of each activity in the sequence is modeled regardless of how far away from prediction the activity occurred and we achieve that through the temporal activity state change representation modeling as described earlier.

The learned activity embeddings and contributions of each activity are then summarized to obtain new activity representation v_{e_i}

$$\forall_{h_{e_i} \in \{i = 1 \ldots l_n\}} \forall_{\delta(e_i, \Delta_t) \in \{i = 1 \ldots l_n\}} v_{e_i} = h_{e_i} \times \delta(e_i, \Delta_t), \tag{8}$$

resulting again in $v_{e_i} \in \mathbb{R}^{d_w = 200}$ dimensional space. For activities modeled in this way, we can easily analyze their initial and time-varying impact during prediction phase.

Recurrent net block. Activity embeddings v_{e_i} are currently sequence oblivious, thus a bidirectional RNN model[42] (with GRU cells[43]) is employed to capture sequence information:

$$g_{e_1}, g_{e_2}, \ldots, g_{e_N} = biRNN(v_{e_1}, v_{e_2}, \ldots, v_{e_N}, \theta_{GRU}). \quad (9)$$

Bidirectional RNN architecture can learn complex relations between activities, including higher order session-level information (activities naturally grouped together).[12] The resulting embeddings g_{e_i} are projected to $\mathbb{R}^{d_m=100}$ dimensional space.

Hard attention block—Discrete representation block. As we discussed in detail, ability of algorithm to handle inherent noise in the data is important for its performance. Existing soft attention approaches are only capable of assigning smaller weights to noisy signals, not being able to remove their potentially harmful effect on algorithms performance. To that end, we employ the hard attention mechanism, proposed for image segmentation,[27,37] to a sequence modeling task. Contrast between soft and hard attention mechanisms is graphically presented in Figure 3a and b.

The process of building hard attention block incorporates discrete neurons that can model zero weights on the existing representations of activities g_{e_i}. As a choice of discrete neurons we experimented with Gumbel-softmax and semantic hashing, both described in detail in Exploring Stochastic Hard Attentions for Modeling Noisy Data section as both have shown merits on different tasks.

Hard attention block learns scores a_{e_i} that are used to reweight embeddings from the RNN layer g_{e_i} as

$$\forall_{g_{e_i} \in \{i=1 \ldots l_n\}} \forall_{a_{e_i} \in \{i=1 \ldots l_n\}} q_{e_i} = g_{e_i} \odot a_{e_i}, \quad (10)$$

resulting in $q_{e_i} \in \mathbb{R}^{d_m=100}$ dimensional space for Gumbel-softmax, and for semantic hashing $q_{e_i} \in \mathbb{R}^{d_m=18}$, as $2^{18} = 262{,}144$ is the first power of 2 larger than the number of unique activities considered in our data sets (see Discretization Through Semantic Hashing section). Operation $\odot$ is an element-wise product.

It should be noted that the difference between the two approaches is that Gumbel-softmax would give a single weight to entire vector g_{e_i} (values are replicated along the d_w dimension), whereas semantic hashing would learn a weight for each dimension of g_{e_i} independently. Semantic hashing thus, thanks to its sigmoid function, has the ability to remove only particular dimensions of activity embeddings rather than removing entire activity representation. This approach has finer granularity of modeling noise contained within different activities, especially for activities that show heterogeneous properties.

Changing Gumbel-softmax to Gumbel-sigmoid to achieve the same provided no improvements in our initial experiments, and it was thus omitted from experiments. The two approaches are compared in detail in the experimental evaluation.

Optimizing with hard attention. Training with discrete neurons (discretization) can be challenging as initial gradient updates have to pass through the discretization bottleneck. The following training procedure is used to address this challenge: for the first 10,000 updates the entire network is trained without this discretization layer, after which the layer is turned on until training is completed. In the first 10,000 steps we

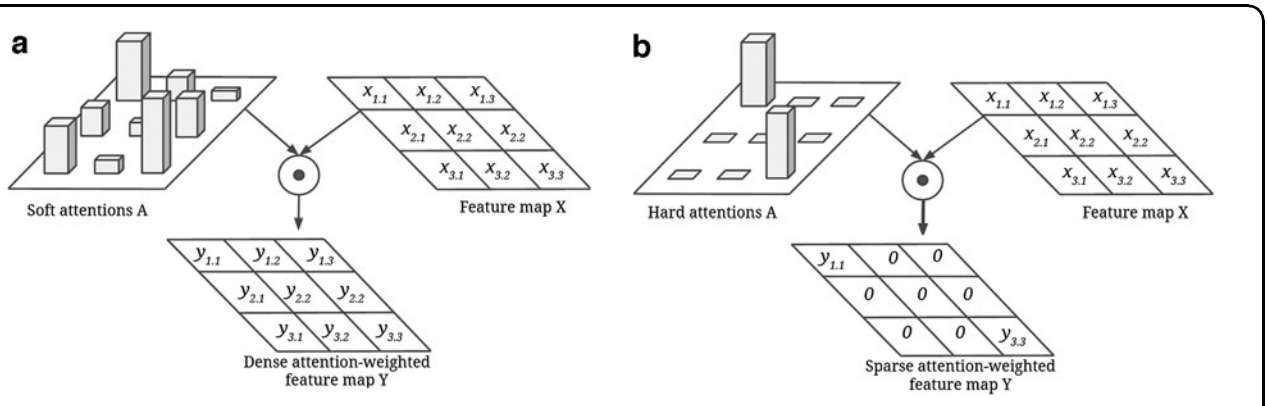

FIG. 3. Comparison between soft and hard attention mechanisms shown through their affect on learned feature maps. **(a)** Soft attention. **(b)** Hard attention.

expect the loss will approach convergence, whereas after switching on discretization layer the loss increases slightly before improving for the remainder of training phase.

Soft attention learning block. Vector summarization of input representation producing matrix is a good practical approach for classification tasks. Attention mechanism was shown to be a good approach for highlighting important parts of the sequence[3,12,44] and using the scores for obtaining weighted input vector representation.

Attention mechanism is trained to provide activity-level scores that highlight activities of greater importance for the task at hand. We implement the attention model through a two-layered neural network $s_q(g_e; \theta_e)$ with softmax outputs and the following dimension: $\mathbb{R}^{d_{a_1}=100}$ and $\mathbb{R}^{d_{a_2}=1}$:

$$t_{e_i} = \frac{\exp(s_e(g_{e_i}; \theta_e))}{\sum_{i=1}^{l_n} \exp(s_e(g_{e_i}; \theta_e))}. \quad (11)$$

The neural network $s_e(g_{e_i}; \theta_e)$ learns real valued scores (attentions) for each ith activity in a given user trail. Obtained attentions t_{e_i} are then used to reweight discretized representation of each activity in sequence q_{e_i} and to obtain compact summarization of the entire sequence $s = \sum_i t_{e_i} * q_{e_i}$ that can be easily used in the following prediction block of the network architecture.

Learning to predict from the resulting representation. Two fully connected layers with inner dimension of $\mathbb{R}^{d_c=100}$ and ReLU nonlinearities with final sigmoid layer $\sigma(\cdot)$ are applied on the summarized vector to obtain the probability of conversion.

For the tasks considered, the standard logistic loss $\mathcal{L}$ was used for optimizing the network parameters:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N} (y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)), \quad (12)$$

where $\hat{y}_n$ are obtained logits after final sigmoid layer and y_n is label for the nth user trail.

Experimental Evaluation

The data used for our experiments are first described in detail, followed by description of baselines and evaluation metrics, and finally, results on described data sets are provided and discussed.

Data description

Experiments are conducted on user activity trails data collected across Yahoo's assets.[†] These include chronologically ordered activities derived from multiplex sources such as Yahoo Search, commercial e-mail receipts, news, and other content reads on Yahoo and AOL homepages, Yahoo Finance, Sports and News, HuffPost, TechCrunch, and so on, advertising data from Yahoo Gemini and Verizon Media DSP (e.g., ad impressions, clicks, conversions, and site visits). Each activity consists of activity ID, time stamp, its type (e.g., search, invoice, reservation, content view, order confirmation, and parcel delivery), and a canonicalized and normalized raw description of the activity (e.g., the search query term for search activities).

All user consensual, advertiser-specific, and local legal guidelines have been considered when creating the data sets.

Data sets used in this study are collected from two anonymized major advertisers from retail and communications domains that we will denote as advertiser A and advertiser B, respectively. Advertiser A has defined three different conversion tasks for its three retail portfolios, whereas advertiser B defined a single conversion task. Training sets for the two advertisers comprised (after eligible users and activities are selected and negatives downsampling is performed to maintain roughly 10% of positives) 1,094,038 trails in train and 273,105 for test set for advertiser A, and of 959,540 trails in train and 239,610 for test set for advertiser B, collected over an undisclosed period >100 days.

For both data sets, it is possible that the same user has multiple trails for multiple conversions; however, this occurs in <1% of users. A common activities vocabulary was selected for the two advertisers, and it contains 243,713 the most prevalent activities. Filtering of activities is done before downsampling negative users so as to select activities that occurred in >1000 unique user trails. The maximum length of user activity trails was selected to be 500 activities after deduplication as per data set statistics ($\sim$80% of all users had sequence length $\leq$500).

Baselines

Models representing previously published studies or models that are expected to fit well with the given setup are used as main baselines in this study:

[†]All data sets used in this study are published for academic use at https://webscope.sandbox.yahoo.com/catalog.php?datatype=a&did=87 (last accessed April 2022).

1. Random forests (RF): RF algorithm with 1000 trees ran on one-hot encoded features from activities in trails. Top 500 features are selected for each task using the chi-squared feature selection. This process mimics the exact setup used in the current production system.
2. Recurrent neural network (RNN): A recurrent neural architecture using efficient GRU cells on top of embedding layer with two fully connected layers for classification.
3. 1D convolutional neural networks (CNN): A one-dimensional temporal convolution on top of learned activity embeddings with two fully connected layers for classification.
4. RNN with attention layer (RNN+Attn): An extension of the RNN model with additional attention layer used to summarize the sequence.[44]
5. RNN with self-attention layer (RNN+SelfAttn): Alternative extension of the RNN model with self-attention layer.[25]

To evaluate the two new blocks, temporal and hard attention, we build on the best performing algorithm from published baselines, which is RNN+Attn in our experiments and shows improving results of RNN with temporal attention (RNN+TimeAttn) and a complete START framework with Gumbel and semantic hashing discretization approaches. The detailed ablation study results are provided in Supplementary Appendix SA1.

Model configuration and training. As the proposed START model is built on top of RNN+Attn baseline, we kept all hyperparameters of RNN-based approaches the same as shown in Methodology section. CNN architecture uses four 1D convolutional blocks with 64 filters of width 3 and batch normalization between layers, the resulting output is flattened and forwarded to the same classification block used by the other algorithms. To optimize $\mathcal{L}$, we use stochastic gradient descent with Adam optimizer, and the best learning rate found through grid search was 0.001, whereas weights are initialized by the truncated normal initializer.

All deep learning models were trained on distributed TensorflowOnSpark[‡] infrastructure with 20 GPU (Nvidia K80) machines.

[‡]https://github.com/yahoo/TensorflowOnSpark (last accessed October 2021).

Evaluation metrics. Quality of estimated classification probabilities is estimated using the AUC classification performance measure with accuracy, precision, and recall scores obtained after choosing the classification threshold.

Experimental results: Combining temporal and hard attentions with existing approaches

In the first phase of experimental evaluation, we assess the effect of modeling temporal signals and using discretization in the hidden layers to model noisy signals. We discuss results on both binary and multiclass classification tasks for the two major advertisers.

Results—Binary classification. In the first step, we rank the baselines by their performance, followed by a discussion on how the proposed extensions fare with the best performing one. The first task is the binary classification where we predict whether a user will convert to any of the conversion tasks set by the two advertisers (Table 1). We observe the difference in predicting conversions for retail versus communications advertisers, where retail category conversion prediction tasks tend to be easier. The heterogeneity of the data is juxtaposed with the target models are optimizing for, and thus for different data sets, we may expect different performances.

Table 1. Performance metrics on binary classification task for advertisers A and B for all baseline models and temporal and hard attention extensions

	ROC AUC	*Accuracy*	*Precision*	*Recall*
Advertiser A				
RF	0.9470	0.8715	0.5715	0.8764
CNN	0.9522	0.9086	0.6719	0.8695
RNN	0.9557	0.9093	0.6717	0.8789
RNN+Attn	*0.9597*	*0.9193*	*0.7031*	*0.8820*
RNN+SelfAttn	0.9544	0.9006	0.6470	0.8719
RNN+TimeAttn	**0.9703**	**0.9376**	**0.7620**	**0.9033**
START(Gumbel)	0.9684	0.9155	0.6832	0.9076
START(SemanticHash)	**0.9772**	**0.9366**	**0.7495**	**0.9236**
Advertiser B				
RF	0.9020	0.7968	0.1564	0.8598
CNN	0.7206	0.6968	0.0862	0.6386
RNN	0.8997	0.8654	0.2113	0.7926
RNN+Attn	*0.9194*	0.8745	0.2279	*0.8169*
RNN+SelfAttn	0.9160	*0.8791*	*0.2326*	0.8017
RNN+TimeAttn	**0.9248**	**0.8960**	**0.2647**	**0.8135**
START(Gumbel)	**0.9267**	0.8519	0.2040	**0.8560**
START(SemanticHash)	**0.9276**	**0.8982**	**0.2696**	0.8162

Italicized values are the best performing baselines and bolded values are novel approaches outperforming them.

AUC, area under the ROC curve; CNN, convolutional neural networks; RF, random forests; RNN, recurrent neural network; RNN+Attn, RNN with attention layer; RNN+SelfAttn, RNN with self-attention layer; ROC, receiver operating characteristic.

RF, a nonsequence modeling approach, shows the lowest performance overall; for advertiser A it is outperformed by all sequence modeling approaches, whereas for advertiser B it is comparable with the simple RNN approach but is outperformed by their attention-based extensions.

As expected according to the available literature,[4,12,15,16] attention-based models seem to stably add value to their baselines, which is especially the case with RNN+Attn approach, whereas RNN+SelfAttn approach does fail to outperform its RNN baseline for the two advertisers, side from accuracy and precision for Advertiser B. Overall, RNN+SelfAttn provides the least improvements over its baseline, whereas adding significant computational overhead as compared with a more traditional RNN+Attn. The latter, without exception, provides improvements across the majority of metrics and is the best performing baseline.

We first experiment with adding temporal modeling layer to the RNN+Attn model[23] as described in Temporal Attention Learning Block section. Results for both advertisers A and B show significant improvement over the best performing baseline yielding the best performing results across the board in Table 1. This confirms our assumptions that a piece of significant information is contained in the temporal aspect of when activities occurred and that sequence of activities only does not contain the complete image of the structure of the data.

We finally discuss results obtained on binary classification task for the two advertisers when using a combination of the performing temporal attention and the hard attention through the two discretization approaches and three strategies (START model). With respect to previously discussed results, the main goal of this analysis is to see if the hard attention mechanism implemented through discretization techniques would provide additional value to the best performing RNN+Attn and RNN+TimeAttn models.

The results using semantic-hashing discretization technique together with time attention truly provided the best performance on both data sets, outperforming both RNN+Attn on all and RNN+TimeAttn models on a majority of metrics. However, Gumbel-softmax discrete approach has not shown stable performance on the two data sets. Even though its performance was consistently better than RNN+Attn model, it only improved performance over RNN+TimeAttn on the second data set. This performance demonstrated that the discrete Gumbel inductive bias may not fit all data well, something which was also observed previously.[38]

Furthermore, it appears that our assumption that through capturing the complete context of the data (in particular the time aspect of activities for our problems) allows the hard attention layer with semantic hashing to truly highlight the remaining noise in the data either through eliminating dimensions of activity representations or eliminating activity entirely, and improving models' overall performance.

Results—Multiclass classification. We further assess the performance of all approaches on multitask classification task (Table 2). This task is defined for advertiser A who specified three different conversion tasks denoted as tasks 2–4, whereas task 1 refers to predicting whether a user will not convert for advertiser A, an opposite task of the binary classification setup.

Similarly, as in the binary classification setup, RF is the worst performing baseline, outperformed by the

Table 2. Performance metrics on multiclass classification task for advertiser A for all baseline models and temporal and hard attention extensions

	ROC AUC	*Accuracy*	*Precision*	*Recall*
Advertiser A—Task 1				
RF	0.9216	0.8294	0.9630	0.8276
CNN	0.9529	0.9020	0.9773	0.9037
RNN	0.9603	0.9086	0.9767	0.9124
RNN+Attn	0.9588	0.9109	0.9761	0.9158
RNN+TimeAttn	**0.9735**	**0.9382**	**0.9811**	**0.9442**
START(Gumbel)	0.9185	0.8419	0.9629	0.8432
START(SemanticHash)	**0.9727**	**0.9362**	**0.9815**	**0.9413**
Advertiser A—Task 2				
RF	0.9143	0.8353	0.0918	0.8075
CNN	0.8605	0.7963	0.0774	0.8346
RNN	0.8811	0.7704	0.0689	0.8310
RNN+Attention	0.9221	0.8062	0.0847	0.8798
RNN+TimeAttn	**0.9423**	**0.8303**	**0.0984**	**0.9098**
START(Gumbel)	0.8740	0.7520	0.0648	0.8425
START(SemanticHash)	**0.9448**	**0.8425**	0.1054	**0.9118**
Advertiser A—Task 3				
RF	0.9102	0.8193	0.2371	0.8100
CNN	0.9130	0.8428	0.2716	0.8516
RNN	0.9246	0.8424	0.2714	0.8539
RNN+Attn	0.9333	0.8512	0.2843	0.8580
RNN+TimeAttn	**0.9494**	**0.8752**	**0.3278**	**0.8853**
START(Gumbel)	0.8898	0.7729	0.2011	0.8457
START(SemanticHash)	**0.9520**	**0.8750**	**0.3290**	**0.8976**
Advertiser A—Task 4				
RF	0.8951	0.8007	0.2597	0.8080
CNN	0.9070	0.8275	0.2982	0.8568
RNN	0.9208	0.8374	0.3122	0.8616
RNN+Attn	0.9247	0.8469	0.3246	0.8495
RNN+SelfAttn	0.9234	0.8423	0.3189	0.8581
RNN+TimeAttn	**0.9448**	**0.8740**	**0.3769**	**0.8851**
START(Gumbel)	0.8825	0.7862	0.2478	0.8237
START(SemanticHash)	**0.9465**	**0.8717**	**0.3737**	**0.8968**

Bolded values are novel approaches outperforming them.
ROC, receiver operating characteristic.

sequence learning approaches CNN and RNN, further strengthening our arguments and literature that sequence of activities contains useful information for predictive tasks. Moreover, the best performing baseline across the board is RNN+Attn yet again, outperforming the self-attention mechanism as well.

RNN with temporal attention mechanism is the better performing approach as compared with baselines on this data set as well, stably outperforming all of the baselines by a significant margin. For the START model, in this experiment, where the same activity representations are shared for multiple predictive task, semantic-hashing hard attention shows the best performance overall, similarly as in the binary classification task. However, in this, more complex experiment, discretization through Gumbel-softmax could not provide additional improvement.

We can finally conclude that once a sufficient amount of patterns in the data are captured through good modeling choices, such as modeling sequence and temporal signals, the discretization layers with appropriate assumptions can indeed help reduce the remainder noise in the data with respect to the task and improve the performance.

Aforementioned experiments show that activities in users' trails need to be given in a context of a sequence with nonuniformly distributed activities where each activity can hold different information depending on the context and the optimization task. Thus, the proposed temporal and hard attention blocks in the START model allow for capturing only useful information from each event in a sequence by mitigating or completely remove the effects of portions of learned embeddings. Analysis of the effects of the two blocks of the START model will be discussed in the following experiments.

Analyzing attentions of different models

Finally, we assess the attention values of different models to provide insights into how models see the data and how do they cope with the noise through different attention mechanisms by analyzing 100 randomly selected user trails that ended with conversion for the two advertisers.

Temporal and soft attention analysis. We first analyze the soft and temporal attention mechanisms. Figure 4a and b shows soft attentions of RNN+Attn model, whereas Figure 4c and d shows soft attentions of START(SemanticHashAttn) model for advertisers A and B, respectively. We can see a very similar pattern where for both algorithms soft attention prefers individual activities that are closer to the end of the trail. However, this artifact is most likely observed due to sequence modeling of the RNN block squeezing most of the information in single vectors toward the end of trails.

For the START model, there are more activities, especially consecutive ones, highlighted by the algorithm, especially for the retail advertiser which is to be expected, whereas for communications advertiser conversions tend to happen close to strong predictive signals. We interpret this difference between models as the proposed blocks help in preserving information from activities by removing the noisy bits through the hard attention block, thus not forcing algorithm to summarize information of sequence in only a few vectors. Next, in Figure 4c, f, g, and h we show θ_{e_i} and μ_{e_i} temporal attention parameters for the two advertisers.

Differently from the aforementioned soft attention, temporal attention, being in the lower parts of the architecture, highlights events across the trail with higher positive values of θ_{e_i} and lower negative values of μ_{e_i}. We can see that there are activities that occur earlier in the trail with significant temporal impact, whereas there are some activities closer to the conversion event with lower impact as well. Observing a similar pattern for all user trails and for both advertisers shows that the temporal attention mechanism properly captures both the long- and short-term temporal impact of each activity before the RNN block that models the sequential process.

Hard attention analysis. We finally show hard attention scores as learned by the START model using the semantic hashing technique. We graphically display them by randomly selecting 9 user activity trails with a conversion for advertiser A, and plotted the hard attention feature map for 500 activities (with null activity padded in the beginning) and for each of the 18 dimensions. The plots are shown in Figure 5 where the values are zeroes or ones as learned by the START model.

We can see that the START algorithm prefers to select certain dimensions over others, sometimes completely deselecting an entire dimension for all activities. This is especially observed for the padded activity. For activities closer to the conversion point, we see more randomness in dimension selection; however, we almost never observe that all of the dimensions of activity are being selected.

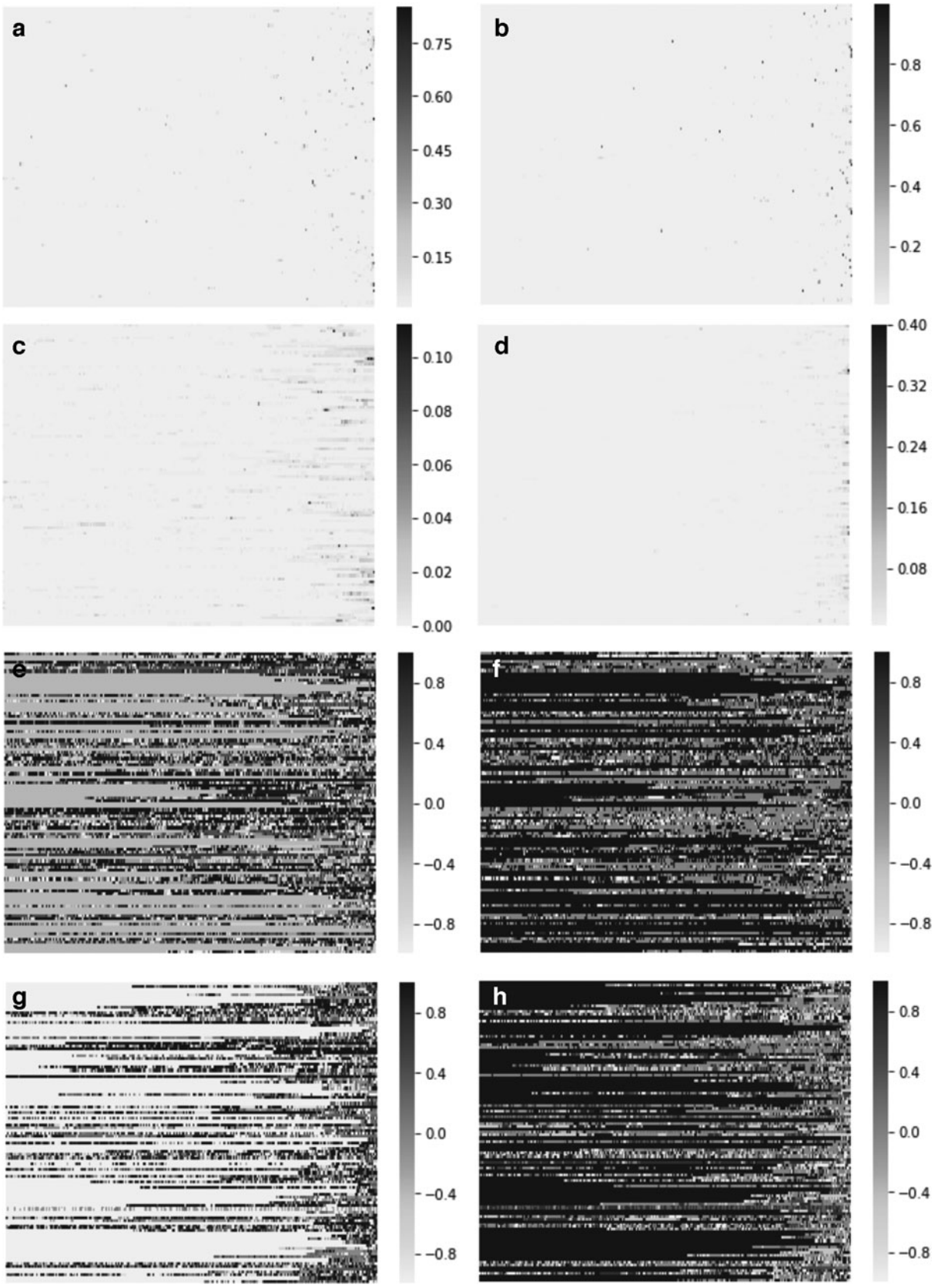

FIG. 4. Various attention score heat maps of events from 100 randomly sampled converters for advertiser A. **(a)** Adv A GRU+Attn attention. **(b)** Adv B GRU+Attn attention. **(c)** Adv A START soft attention. **(d)** Adv B START soft attention. **(e)** Adv A START θ_{e_i}. **(f)** Adv A START μ_{e_i}. **(g)** Adv B START θ_{e_i}. **(h)** Adv B START μ_{e_i}.

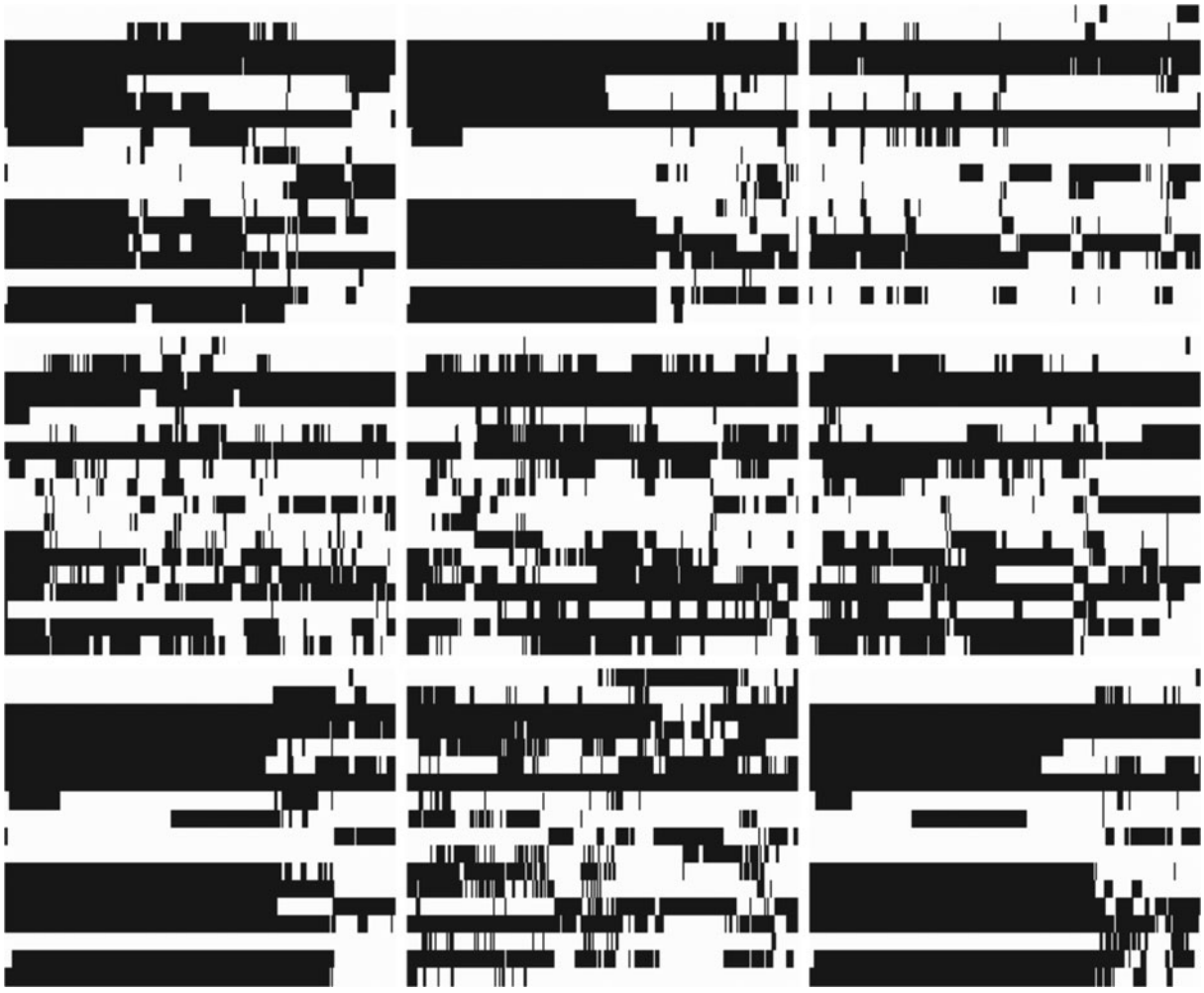

FIG. 5. START models semantic hashing hard attention examples of eight user trails. Columns are 18 dimensional representations of 500 events in the activity trail length. Blue and yellow cells represent ones and zeros assigned to those locations in the user trail feature map.

Selecting one or a few dimensions from the activity embedding means that the algorithm "learned" which parts of the observed activity signal is particularly useful in the context of the observed user activity trail, as opposed to using the entire representation and potentially including dimensions that may not be useful for prediction task or maybe just noise. Nonselected dimensions of one activity can be active for another user's trail.

Activities collected online often contain less information when taken out of the context of the sequence they were found in; moreover, the same activity can carry different information for different sequences and for different optimization tasks. Having a mechanism that is capable of decomposing the available signal and using only its important parts has a great impact in annealing the inherent noise in heterogeneous signals, ultimately improving the generalization power of an algorithm. This is the exact benefit of using hard attention for dealing with noisy activities data collected from multiplex sources.

Conclusions

In this study, we tackled the problems of modeling user conversion probabilities for online advertising given activity inputs collected across many different sources

ordered in time. Major challenges tackled were (1) learning to account for time information as collected activities do not uniformly occur and (2) learning to filter out inherent noise in data.

To that end, we exploited the temporal attention mechanism that learns both time-invariant and time-dependent effects of each activity and proposed the hard attention mechanism through discrete layers for annealing noise inherent in the data that can be learned in the network using standard optimization techniques. The two extensions of existing sequence modeling approach allowed us to obtain significant improvements across several metrics without any major drawbacks, thus providing a step forward in the improvement of the overall conversion prediction systems.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

No funding was received for this article.

Supplementary Material

Supplementary Appendix SA1

References

1. Gligorijevic D, Zhou T, Bharatbhushan S, et al. Bid shading in the brave new world of first-price auctions. In: d'Aquin M, Dietze A, Hauff C, et al. (Eds.): Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19–23, 2020. pp. 2453–2460.
2. He X, Pan J, Jin O, et al. Practical lessons from predicting clicks on ads at Facebook. In: Saka E, Shed D, Lee K, et al. (Eds.): Proceedings of the 8th International Workshop on Data Mining for Online Advertising, AdKDD 2014. ACM, New York, NY, August 24–27, 2014, pp. 1–9.
3. Gligorijevic D, Stojanovic J, Raghuveer A, et al. Modeling mobile user actions for purchase recommendations using deep memory networks. In: Collins-Thompson K, Mei Q, Davison B, et al. (Eds.): Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018. pp. 1021–1024.
4. Zhou Y, Mishra S, Gligorijevic J, et al. Understanding consumer journey using attention based recurrent neural networks. In: Teredesai A, Kumar V, Li Y, et al. (Eds.): Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, August 4–8, 2019. pp. 3102–3111.
5. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Bengio Y, LeCun Y, Kingsbury B, et al. (Eds.): Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, May 7–9, 2015.
6. Karlsson N. Control problems in online advertising and benefits of randomized bidding strategies. Eur J Control. 2016;30:31–49.
7. Bhamidipati N, Kant R, Mishra S. A large scale prediction engine for app install clicks and conversions. In: Lim E, Winslett M, Sanderson M, et al. (Eds.): Proceedings of the 26th International Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 6–10, 2017. pp. 167–175.
8. Richardson M, Dominowska E, Ragno R. Predicting clicks: Estimating the click-through rate for new ads. In: Williamson C, Zurko ME, Patel-Scheider P, et al. (Eds.): Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8–12, 2007, pp. 521–530.
9. McMahan HB, Holt G, Sculley D, et al. Ad click prediction: A view from the trenches. In: Ghani R, Senator T, Bradley P, et al. (Eds.): Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, August 11–14, 2013. pp. 1222–1230.
10. Pan J, Mao Y, Ruiz AL, et al. Predicting different types of conversions with multi-task learning in online advertising. In: Teredesai A, Kumar V, Li Y, et al. (Eds.): Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, August 4–8, 2019. pp. 1834–1842.
11. Shan Y, Hoens TR, Jiao J, et al. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In: Krishnapuram B, Shah M, Aggrawal C, et al. (Eds.): Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, San Francisco, CL, August 13–17, 2016. pp. 255–262.
12. Gligorijevic J, Gligorijevic D, Stojkovic I, et al. Deeply supervised model for click-through rate prediction in sponsored search. Data Min Knowl Disc. 2019;33:1446–1467.
13. Jiang Z. Research on ctr prediction for contextual advertising based on deep architecture model. J Control Eng Appl Inform. 2016;180:11–19.
14. Zhang Y, Dai H, Xu C, et al. Sequential click prediction for sponsored search with recurrent neural networks. In: Brodley C, Stone P (Eds.): Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec City, Quebec, Canada, July 27–31, 2014.
15. Cui Y, Tobossi R, Vigouroux O. Modelling customer online behaviours with neural networks: Applications to conversion prediction and advertising retargeting. arXiv preprint arXiv:1804.07669, 2018.
16. Arava SK, Dong C, Yan Z, et al. Deep neural net with attention for multichannel multi-touch attribution. arXiv preprint arXiv:1809.02230, 2018.
17. Pei W, Tax DMJ. Unsupervised learning of sequence representations by autoencoders. arXiv preprint arXiv:1804.00946, 2018.
18. Beutel A, Covington P, Jain S, et al. Latent cross: Making use of context in recurrent recommender systems. In: Chang Y, Zhai C, Liu Y, et al. (Eds.): Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018. ACM, Marina Del Rey, CA, February 5–9, 2018. pp. 46–54.
19. Li Y, Du N, Bengio S. Time-dependent representation for neural event sequence prediction. In: Bengio Y, LeCun Y, Sainath T, et al. (Eds.): 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Workshop Track Proceedings.
20. Jing H, Smola AJ. Neural survival recommender. In: de Rijke M, Shokouhi M, Tomkins A, et al. (Eds.): Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6–10, 2017. pp. 515–524.
21. Zhu Y, Li H, Liao Y, et al. What to do next: Modeling user behaviors by time-LSTM. In: Sierra C (Eds.): Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017. pp. 3602–3608.
22. Rajkomar A, Oren E, Chen K, et al. Scalable and accurate deep learning with electronic health records. NPJ Digital Med. 2018;1:18.
23. Gligorijevic D, Gligorijevic J, Flores A. Prospective modeling of users for online display advertising via deep time-aware model. In: d'Aquin M, Dietze A, Hauff C, et al. (Eds.): Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19–23, 2020. pp. 2461–2468.
24. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Ghahramani Z, Welling M, Cortes C, et al. (Eds.): Proceedings of the 27th International Conference on Neural Information Processing Systems, NeurIPS 2014, Montreal, Canada, December 8–13 2014. pp. 3104–3112.
25. Vaswani A, ShazeerN, Parmar N, et al. Attention is all you need. In: von Luxburg U, Guyon I, Bengio S, et al. (Eds.): Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS 2017, Long Beach, CA, December 4–9, 2017. pp. 5998–6008.
26. Gligorijevic D, Stojanovic J, Satz W, et al. Deep attention model for triage of emergency department patients. In: Obradovic Z, Parthasarathy S, Apte C, et al. (Eds.): Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018, San Diego, CA, May 3–5, 2018. pp. 297–305.

27. Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention. In: Bach F, Blei D (Eds.): Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML 2015, Lille, France, July 6–11, 2015. pp. 2048–2057.
28. Metz L, Ibarz J, Jaitly N, Davidson J. Discrete sequential prediction of continuous actions for deep RL. arXiv preprint arXiv:1705.05035, 2017.
29. Kaiser q, Bengio S. Discrete autoencoders for sequence models. arXiv preprint arXiv:1801.09797, 2018.
30. Bengio Y, Leonard N, Courville A. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013.
31. Gulcehre C, Chandar S, Bengio Y. Memory augmented neural networks with wormhole connections. arXiv preprint arXiv:1701.08718, 2017.
32. Kingma DP, Welling M. Auto-encoding Variational Bayes. arXiv preprint arXiv:1312.6114, 2013.
33. Maddison CJ, Mnih A, Teh YW. The concrete distribution: A continuous relaxation of discrete random variables. In: Bengio Y, LeCum Y, Ranzato M, et al. (Eds.): Proceedings of 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017.
34. Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-Softmax. In: Bengio Y, LeCum Y, Ranzato M, et al. (Eds.): Proceedings of 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017.
35. Salakhutdinov R, Hinton G. Semantic hashing. Int J Approx Reason. 2009; 50:969–978.
36. Shen C, Qi G-J, Jiang R, et al. Sharp attention network via adaptive sampling for person re-identification. IEEE Trans Circuits Syst Video Technol. 2019;29:3016–3027.
37. Yan S, Smith JS, Lu W, Zhang B. Hierarchical multi-scale attention networks for action recognition. Signal Process Image Commun. 2018;61: 73–84.
38. Kaiser L, Bengio S. Can active memory replace attention? In: Lee D, von Luxburg U, Garnett R, et al. (Eds.) Proceedings of the 30th International Conference on Neural Information Processing Systems, NeurIPS 2016, Barcelona, Spain, December 5–10, 2016. pp. 3781–3789.
39. Louizos C, Welling M, Kingma DP. Learning sparse neural networks through l 0 regularization. In: Bengio Y, LeCun Y, Sainath T, et al. (Eds.): 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018.
40. Cao XH, Han C, Obradovic Z. Learning a dynamic-based representation for multivariate biomarker time series classifications. In: 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018, IEEE, 2018. pp. 163–173.
41. Bai T, Zhang S, Egleston BL, Vucetic S. Interpretable representation learning for healthcare via capturing disease progression through time. In: Guo Y, Farooq (eds.): Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, United Kingdom, August 19–23, 2018. pp. 43–51.
42. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;450:2673–2681.
43. Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Moschitti A, Pang B, Alfonseca E, et al. (Eds.): Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 25–29, 2014. pp. 1724–1734.
44. Zhai S, Chang K-H, Zhang R, Zhang ZM. Deepintent: Learning attentions for online advertising with recurrent neural networks. In: Krishnapuram B, Shah M, Aggrawal C, et al. (Eds.): Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, San Francisco, California, August 13–17, 2016. pp. 1295–1304.

Cite this article as: Gligorijevic D, Gligorijevic J, Flores A (2022) Predicting actions of users using heterogeneous online signals. *Big Data* 3:X, 1–15, DOI: 10.1089/big.2021.0320.

Abbreviations Used

ads = advertisements
AUC = area under the ROC curve
CNN = convolutional neural networks
DA = display advertising
RF = random forests
RNN = recurrent neural network
RNN+Attn = RNN with attention layer
RNN+SelfAttn = RNN with self-attention layer
ROC = receiver operating characteristic