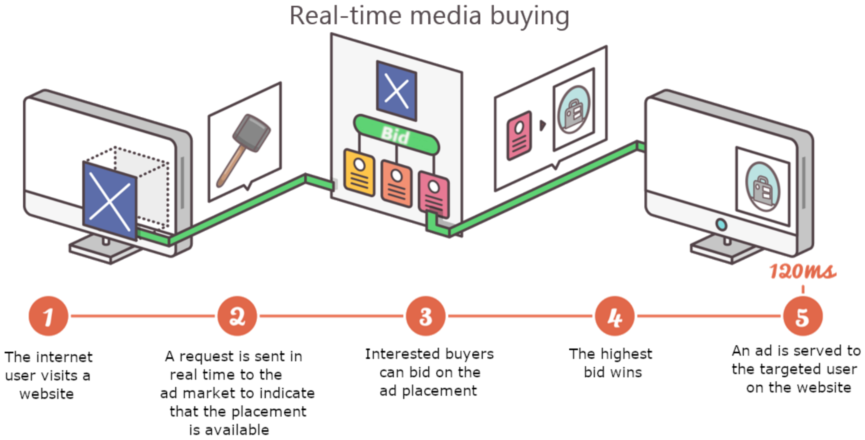
Predicting conversions in display advertising based on URL embeddings

Yang QIU [1,2], Nikolaos Tziortziotis[1], Martial Hue[1], Michalis Vazirgiannis[2]

[1] Jellyfish
yanq.qiu@jellyfish.com, ntziorzi@gmail.com, martial.hue@jellyfish.com
[2] LIX, École Polytechnique
mvazirg@lix.polytechnique.fr

August 23, 2020

A high-level overview of RTB procedure



Motivation

Motivation

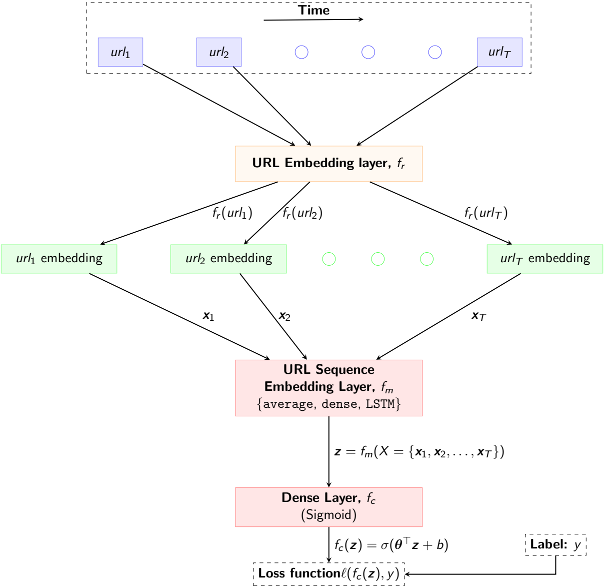
- Accurate CVR (Conversion Rate) estimation is essential to effectiveness of campaigns
- User's browsing history is rarely explored but contains plenty of information
- URL's cardinality is too high to let URL be used directly in the model

Inspired by NLP techniques, we aim to learn an embedding of URL in a self supervised way and use it to improve performance of CVR prediction task.

Objectives (conversion prediction)

Given user's browsing history (sequence of URLs visited by user) on day $\mathcal{D}_d$, predict if user will convert or not to a specific advertiser on day $\mathcal{D}_d + 1$

Proposed conversion prediction model architecture



Proposed conversion prediction model architecture

The model can be divided into two learning phases

1. URL embedding learning phase:
   train f_r via URL representation model using non labeled user sequences

2. Conversion prediction learning phase:
   freeze f_r, train f_m and f_c using labeled user sequences

URL representation layer f_r

$f_r : url \rightarrow \boldsymbol{x} \in \mathbb{R}^d$ where url = sequence of tokens (truncate to maximum 3 tokens)

One-hot encoding

$$f_r(url) = \frac{1}{n\_tokens} \sum_{t=1}^{n\_tokens} \boldsymbol{e}_t,$$

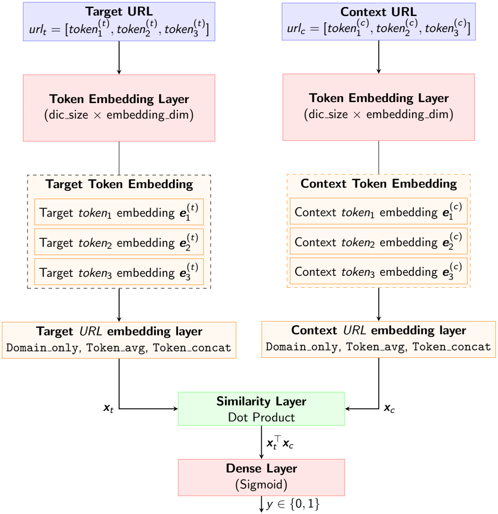
where $\{\boldsymbol{e}_t\}_{t=1}^{n\_tokens \leq 3}$ is a one-hot encoding $\forall$ token $\in url$.

URL embedding

Given token embedding vectors $\{\boldsymbol{e}_t\}_{t=1}^{3}$:

- `Domain_only`: $f_r(url) = \boldsymbol{e}_1$
- `Token_avg`: $f_r(url) = \frac{1}{3} \sum_{t=1}^{3} \boldsymbol{e}_t$
- `Token_concat`: $f_r(url) = [\boldsymbol{e}_1^\top, \boldsymbol{e}_2^\top, \boldsymbol{e}_3^\top]^\top$

URL representation model (skip-gram[1])



[1] T. Mikolov, et. al., Distributed Representations of Words and Phrases and their Compositionality, NIPS, 2013.

URL sequence representation layer f_m

$f_m : X \to \boldsymbol{z} \in \mathbb{R}^m$ where $X = [x_1, \ldots, x_T]$

Three possible mapping functions:

- `average` (LR):

$$f_m^{(1)}(X) = \frac{1}{T} \sum_{i=1}^{T} \boldsymbol{x}_i$$

- `dense` (DLR):

$$f_m^{(2)}(X) = g(\boldsymbol{\theta}^{(1)\top} f_m^{(1)}(X) + \boldsymbol{b}^{(1)}),$$

  where g is the ReLU activation function.

- `LSTM` (RNN) [2]: $f_m^{(3)}(X) = LSTM(X)$

[2] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory, 1997

Dataset

- We used a real RTB dataset collected from our campaigns launched in France
- Each record of the dataset corresponds to a chronologically ordered sequence of visited URLs
- The maximum length of a URL sequence is set equal to 500, while keeping the lastest URLs if exceeds
- Tokens appeared less than 20 times considered "rare"
- We use the data of 3 consecutive days:
  - train the URL representation model using data on day $\mathcal{D}_d$
  - train the conversion prediction model using data on day $\mathcal{D}_d + 1$, label is defined based on conversions on day $\mathcal{D}_d + 2$
  - test the conversion prediction model using data on day $\mathcal{D}_d + 2$, label is defined based on conversions on day $\mathcal{D}_d + 3$

Empirical analysis

In total we examine ten prediction models:

One_hot/LR	Domain_only/LR	Token_avg/LR	Token_concat/LR
	Domain_only/DLR	Token_avg/DLR	Token_concat/DLR
	Domain_only/RNN	Token_avg/RNN	Token_concat/RNN

Conversion prediction model training and testing data

Advertiser Category	**Training** (5,452,577)	**Testing** (7,164,109)
Banking	(3,746 – 5,448,831)	(8,539 – 7,155,570)
E-shop	(1,463 – 5,451,114)	(1,821 – 7,162,288)
Newspaper_1	(1,406 – 5,451,171)	(2,923 – 7,161,186)
Newspaper_2	(1,261 – 5,451,316)	(1,291 – 7,162,818)
Telecom	(1,781 – 5,450,796)	(2,201 – 7,161,908)

Number of converted vs. non-converted records

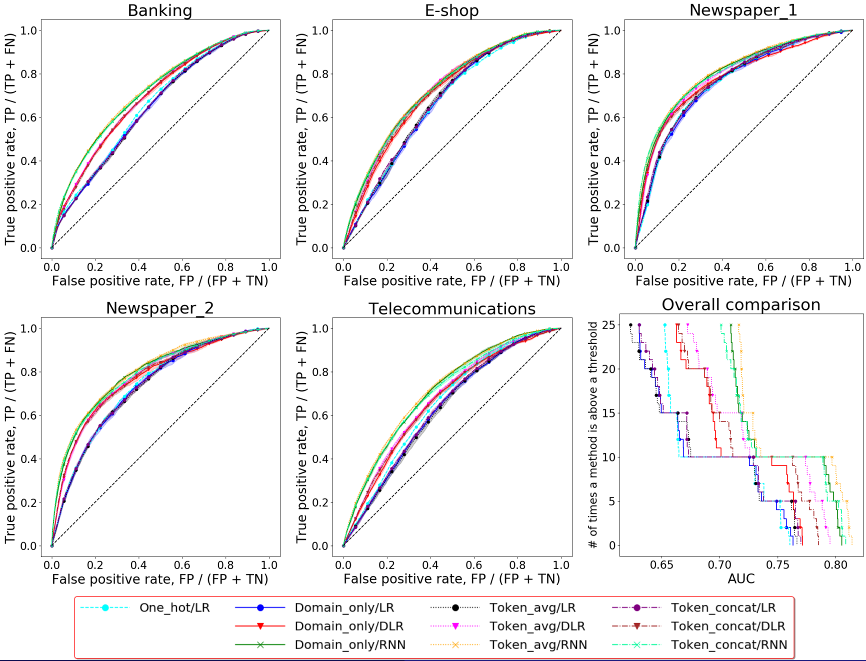
URL embedding visualisation

Figure: t-SNE visualization of the thirty closest neighbors of 24 different domains.

Conversion prediction performance comparison

Method \ Adv	**Banking**	**E-shop**	**Newspaper_1**	**Newspaper_2**	**Telecom**
One_hot/LR	65.7 ± 0.093	66.4 ± 0.053	75.5 ± 0.379	73.3 ± 0.400	65.4 ± 0.085
Domain_only/LR	64.6 ± 0.300	66.6 ± 0.217	75.7 ± 0.507	73.2 ± 0.345	63.2 ± 0.168
Domain_only/DLR	69.0 ± 0.214	69.7 ± 0.234	76.8 ± 0.342	75.7 ± 0.658	66.6 ± 0.303
Domain_only/RNN	71.4 ± 0.144	72.6 ± 0.422	80.3 ± 0.168	79.4 ± 0.281	71.2 ± 0.250
Token_avg/LR	64.5 ± 0.241	67.2 ± 0.390	76.4 ± 0.152	73.1 ± 0.184	62.9 ± 0.468
Token_avg/DLR	69.4 ± 0.294	72.1 ± 0.263	79.2 ± 0.242	77.7 ± 0.274	67.9 ± 0.348
Token_avg/RNN	$\mathbf{71.9} \pm 0.082$	$\mathbf{73.1} \pm 0.246$	$\mathbf{81.2 \pm 0.153}$	$\mathbf{80.2 \pm 0.322}$	$\mathbf{71.8 \pm 0.153}$
Token_concat/LR	64.8 ± 0.241	67.2 ± 0.060	76.7 ± 0.179	73.4 ± 0.273	63.6 ± 0.425
Token_concat/DLR	69.1 ± 0.222	70.8 ± 0.400	78.2 ± 0.285	76.7 ± 0.255	66.9 ± 0.310
Token_concat/RNN	71.5 ± 0.224	72.5 ± 0.460	80.5 ± 0.192	79.1 ± 0.130	70.5 ± 0.278

Table: Avg (%) and std of the AUC score (5 independent runs) of the 10 prediction models on 5 advertisers.


Banking
E-shop
Newspaper_1
Newspaper_2
Telecommunications
Overall comparison
True positive rate, TP / (TP + FN)
False positive rate, FP / (FP + TN)
# of times a method is above a threshold
AUC
One_hot/LR
Domain_only/LR
Domain_only/DLR
Domain_only/RNN
Token_avg/LR
Token_avg/DLR
Token_avg/RNN
Token_concat/LR
Token_concat/DLR
Token_concat/RNN

Conclusions

- all three proposed URL embedding models are able to learn high-quality vector representations that capture the URL relationships
- the performance of the `LR` model is relatively invariant to the selection of the representation model
- `Token_avg` is more adequate to capture the relationship between URLs, with the `Token_concat` second best
- `DLR` and `RNN` both outperform `LR`, with `RNN` dominates the others

Future directions

- add more baseline models' results
- examine the hyperparameters impact (i.e. dimension of embedding vector, etc.) on the performance of our model
- extend our empirical analysis to a real-world online scenario

Thank you for your attention!