Weekly Meeting

Zicun Cong

August 8th, 2016

Problem Definition

Navigation Behavior

An *navigation behavior* B_k is a triple (d, u, t), where d is a device id, u is an URL and t is the time the URL u is visited by the device d.

Navigation Behavior Sequence

A *navigation behavior sequence* $\mathbb{B}$ is a sequence of *navigation behaviors*, $\mathbb{B} = \{B_1, \ldots, B_n\}$, where $B_1.t < B_2.t < \cdots < B_m.t$, $B_1.d = B_2.d = \cdots = B_m.d$, and $B_n.t - B_1.t \leq w$. w is a user defined time window.

Problem Definition

Navigation Behavior Normality

By assuming the navigation behaviors of a device depend on his previous behaviors, we measure the normality of a *navigation behavior* B_k by the conditional probability of B_k given the historical behaviors of the device $B_k.d$ within a short period w, that is,

$$nor(B_k) = P(B_k.u | B_1.u, \ldots, B_{k-1}.u)$$

$B_1, \ldots, B_k$ have the same device ids and $B_k.t - B_1.t \leq w$.

Navigation Behavior Set Normality

The *normality of a navigation behavior sequence*, $\mathbb{B} = \{B_1, \ldots, B_n\}$, is defined as the probability it is generated.

$$nor(\mathbb{B}) = P(\mathbb{B}) = \prod nor(B_k) = \prod P(B_k.u | B_1.u, \ldots, B_{k-1}.u)$$

Proposed Method

- Embed URL
  - Character-level Convolutional Neural Network
- Learn the distribution of behavior sequence
  - Recurrent Neural Network

Character-level Convolutional Neural Network

1. Let C be the vocabulary of characters.
2. The word $k \in V$ is made up of a sequence of characters $\{c_1, \ldots, c_l\}$.
3. The character-level representation of k is given by the matrix $C^k \in \mathbb{R}^{\|C\| * l}$. C^k are zero-padded so that the number of columns of different words are constant.
4. C^k is mapped to a vector $\mathbf{y}^k$ by a convolutional neural network, where $\mathbf{y}^k = \{y_1^k, \ldots, y_h^k\}$. h is the number of filters (or kernels).

Recurrent Neural Network Language Model

We treat each *navigation behavior* as a word and each *navigation behavior sequence* as sentence. We use RNN-LM to learn a distribution over url_{t+1} given historical url sequences $url_1, \ldots, url_t$.

1. Given a time window w, the data are grouped into multiple *navigation behavior sets*.
2. $\mathbf{P}(url_{t+1} = j | url_1, \ldots, url_t) = \frac{exp(\mathbf{h}_t * \mathbf{y}^j)}{\sum_{j' \in URL} exp(\mathbf{h}_t * \mathbf{y}^{j'})}$
3. Given *navigation behavior sets*, RNN tries to minimizing the negative log-likelihood.
   $NLL = -\sum_{n=1}^{N} \sum_{t=1}^{T_n} log\ \mathbf{P}(url_t^n | url_1^n, \ldots, url_{t-1}^n)$
   - N is the number of *navigation behavior sets*
   - T_n is the size of the n-th *navigation behavior set*
   - url_t^n is the t-th url in the n-th *navigation behavior set*