Modeling User Activities on the Web using Paragraph Vector

Yukihiro Tagami, Hayato Kobayashi, Shingo Ono, Akira Tajima
Yahoo Japan Corporation
Tokyo, Japan
{yutagami, hakobaya, shiono, atajima}@yahoo-corp.jp

ABSTRACT

Modeling user activities on the Web is a key problem for various Web services, such as news article recommendation and ad click prediction. In this paper, we propose an approach that summarizes each sequence of user activities using the Paragraph Vector [3], considering users and activities as paragraphs and words, respectively. The learned user representations are used among the user-related prediction tasks in common. We evaluate this approach on two data sets based on logs from Web services of Yahoo! JAPAN. Experimental results demonstrate the effectiveness of our proposed methods.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Social and behavioral sciences**]: Economics

Keywords

Paragraph Vector, Web browsing behavior, representation learning.

1. INTRODUCTION

Large-scale Web sites that provide various Web services deal with a lot of user-related prediction tasks, such as news article recommendation and ad click prediction. Some of these tasks have a few training data, whereas logs of the user activities on the whole Web site are sufficiently available. For such cases, informative user representations, obtained via unsupervised learning with the history of user activities, are useful as features for the prediction tasks.

We propose here an approach that summarizes each sequence of user activities using the Paragraph Vector [3], which is an unsupervised method that learns continuous distributed vector representations from pieces of text. In other words, we apply the vector model to sequences of user activities, considering users and activities as paragraphs (or documents) and words, respectively. The learned low-dimensional feature vectors are used among the user-related prediction tasks in common. We evaluate this approach on two data sets based on logs from Web services of Yahoo! JAPAN. Experimental results demonstrate the effectiveness of our proposed methods.

WWW'15 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742712.

2. METHOD

We briefly describe Paragraph Vector [3] for our setting. A sequence of an i-th user u_i's activities on the Web is represented as $(a_{i,1}, a_{i,2}, \ldots, a_{i,T_i})$, where $a_{i,j}$ is a j-th activity of user u_i and T_i is the size of this sequence. These activities can include Web page visits, search queries, and ad clicks. The objective of the vector model for this sequence is to maximize the average log probability:

$$\frac{1}{T_i}\sum_{t=1}^{T_i} \log p(a_{i,t} \mid a_{i,t-1}, \ldots, a_{i,t-s}, u_i),$$

where s is the size of the context window. The PV-DM, Distributed Memory Model of Paragraph Vectors, defines the probability of this multi-class problem using the softmax function as follows:

$$p(a_{i,t} \mid a_{i,t-1}, \ldots, a_{i,t-s}, u_i) := \frac{\exp(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I)}{\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_I)}, \quad (1)$$

where $\boldsymbol{w}_{a_{i,t}}$ is the output vector corresponding to $a_{i,t}$ and $\boldsymbol{v}_I$ is the concatenated input vector of the previous activities vectors $\boldsymbol{v}_{a_{i,t-1}}, \ldots, \boldsymbol{v}_{a_{i,t-s}}$ and user input vector $\boldsymbol{v}_{u_i}$, that is $\boldsymbol{v}_I = [\boldsymbol{v}_{a_{i,t-1}}^{\mathrm{T}}, \ldots, \boldsymbol{v}_{a_{i,t-s}}^{\mathrm{T}}, \boldsymbol{v}_{u_i}^{\mathrm{T}}]^{\mathrm{T}}$. A is a set of possible user activities. For the case of $j \leq 0$, an input activity vector $\boldsymbol{v}_{a_{i,j}}$ is replaced with a special padding vector $\boldsymbol{v}_{NULL}$. We define the size of input activity vector $|\boldsymbol{v}_{a_{i,j}}|$ as v_a and the size of input user vector $|\boldsymbol{v}_{u_i}|$ as v_u, so the size of both input vector $\boldsymbol{v}_I$ and output vector $\boldsymbol{w}_{a_{i,j}}$ is $s \times v_a + v_u$. The user vector $\boldsymbol{v}_{u_i}$ is used as a feature vector of various prediction tasks, such as news article recommendation and ad click prediction.

The computation of Eq. (1) and its first derivative is impractical because the number of unique activities $|A|$ is typically large. Le and Mikolov [3] originally used hierarchical softmax with a Huffman binary tree based on word frequencies for fast training. Here, instead of hierarchical softmax, we employ negative sampling approach [4]. Hence an alternate objective to $\log p(a_{i,t} \mid a_{i,t-1}, \ldots, a_{i,t-s}, u_i)$ with Eq. (1) is defined as:

$$\log \sigma(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I) + k \cdot \mathbb{E}_{a_n \sim p_n(a)} \left[\log \sigma(-\boldsymbol{w}_{a_n}^{\mathrm{T}} \boldsymbol{v}_I)\right],$$

where $\sigma(z) = 1/(1+\exp(-z))$ is a sigmoid function, k is the number of randomly sampled negative instances, and $p_n(a)$ is a noise distribution generating negative instances. We use the "unigram" distribution $U(a)$ raised to the 3/4rd power as $p_n(a)$ in the same way that Mikolov et al. did [4]. We train the model using asynchronous Stochastic Gradient Descent (SGD) [5] with AdaGrad [2]. In inference step for

Table 1: Experimental results. Values are AUC.

	AdClicker					*SiteVisitor*				
	Ac1	Ac2	Ac3	Ac4	Ac5	Sv1	Sv2	Sv3	Sv4	Sv5
Bin	0.9757	0.7962	**0.6614**	0.7024	0.7476	0.7596	0.8165	0.7080	0.7930	0.7286
Freq	0.9814	0.8068	0.6542	0.6910	0.7433	0.7813	0.8132	0.6977	0.7805	0.7214
Skip-gram	0.9905	0.8337	0.6545	0.7155	0.7710	0.8012	0.8328	0.7129	0.7927	0.7405
PV-DM	0.9900	0.8174	0.6538	0.7303	0.7675	0.8039	0.8356	0.7169	0.7953	0.7462
PV-DM+Skip-gram	**0.9912**	**0.8360**	0.6612	**0.7412**	**0.7758**	**0.8124**	**0.8395**	**0.7248**	**0.8015**	**0.7516**

new users, the user vectors $\boldsymbol{v}_u$ are learned while input and output activity vectors $\boldsymbol{v}_a$ and $\boldsymbol{w}_a$ are fixed.

3. EXPERIMENT

Data sets. We evaluated the proposed method using two supervised learning data sets: *AdClicker* and *SiteVisitor*. *AdClicker* consists of the users who clicked contextual ads that are included in the five selected ad campaigns (Ac1–Ac5). Similarly, *SiteVisitor* consists of the users who visited Web sites of five selected advertisers (Sv1–Sv5).

For simplicity, we created these two data sets in view of predicting a user's particular activities on a day on the basis of the history of Web pages visited the previous day. The training and validation sets were generated from logs of July 22 and 23, 2014. Web page visits in the former day are used as features, and the target activity in the latter day is treated as a label. Similarly, test set was generated from logs of July 23 and 24, 2014, as features and a label, respectively. Since these features were extracted from Web service logs of Yahoo! JAPAN, they are only a small fraction of the entire user activities on the Web. These features do not include visits to the advertiser's site, which is the label of *SiteVisitor*.

Contextual ads in *AdClicker* are determined to be displayed by the Web page content as well as user information. Therefore, learning each Web page representation is also helpful for this task. On the other hand, *SiteVisitor* is the data set based on more complicated user interests.

These two data sets are multi-label data sets because a user can click more than one ad or visit various advertisers' sites. In the experiment, we transformed the multi-label problem into a set of binary classification problems and trained logistic regression classifiers using features extracted by each method. The evaluation measure is Area Under ROC Curve (AUC). The statistics for both data sets are summarized in Table 2.

Table 2: Statistics for two data sets. #Features is the number of unique URLs in each data set.

Data set	#Train	#Validation	#Test	#Features
AdClicker	51,576	10,000	10,000	786,467
SiteVisitor	1,862,693	20,000	20,000	17,574,741

Evaluation setting. We trained PV-DM model with a part of logs of July 22, 2014 and the learned user vectors were used as features for the supervised learning task. We discarded URLs that occurred fewer than five times in the extracted data. We also inserted a special symbol into sequences if an interval of time between two consecutive page visits is greater than 30 minutes. Consequently, the number of unique URLs is about 3.52 million, and about one billion page visits are in the data. The settings of PV-DM learning are as follows: the size of input vectors $v_a = v_u = 400$, the size of context window $s = 5$, the number of randomly sampled negative instances $k = 5$, and the number of epochs (full pass through the data) is five. After training the model, we create the feature vectors via the inference step.

Proposed methods and baselines. We compared the methods using Paragraph Vector with some baselines. *Bin* and *Freq* are weak baselines that use raw URLs as features. *Freq* takes into account of the frequencies of the user's site visits, whereas *Bin* considers only whether a user visits the Web page or not. *Skip-gram* is a method of using vectors learned by the continuous Skip-gram model [4]. This approach represents a user as the simple averaging of activity vectors in the sequence, which is similar to the approach of Djuric et al. [1]. The data and settings of training the Skip-gram model are the same as those of PV-DM. The proposed method using the PV-DM model is represented as *PV-DM*. We also evaluated a method that uses the concatenated vectors learned by the PV-DM and Skip-gram model. This method is called *PV-DM+Skip-gram*. Because of stochastic behavior of asynchronous SGD and random initialization, we report the mean value of five runnings for *PV-DM*, *Skip-gram* and *PV-DM+Skip-gram*.

Results. The experimental results are summarized in Table 1. The **bold** elements indicate the best performance of the methods. The underlined scores are the better results of the *PV-DM* and *Skip-gram*.

PV-DM achieved better results than *Skip-gram* in *SiteVisitor* whereas the opposite trend is shown in *AdClicker*. This is caused by the difference of two data sets as described above. In addition, since the combination method *PV-DM+Skip-gram* performed better than individual methods, these two models learned slightly different but complementary aspects of user activities. Two weak baselines *Bin* and *Freq*, which use raw URLs as features, perform poorly for almost all cases.

These results show the effectiveness of the approach using Paragraph Vector for tasks in other domain than natural language processing.

4. REFERENCES

[1] N. Djuric, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hidden conditional random fields with distributed user embeddings for ad targeting. In *ICDM*, 2014.

[2] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12, 2011.

[3] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[5] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.