Modeling User Activities on the Web using Paragraph Vector

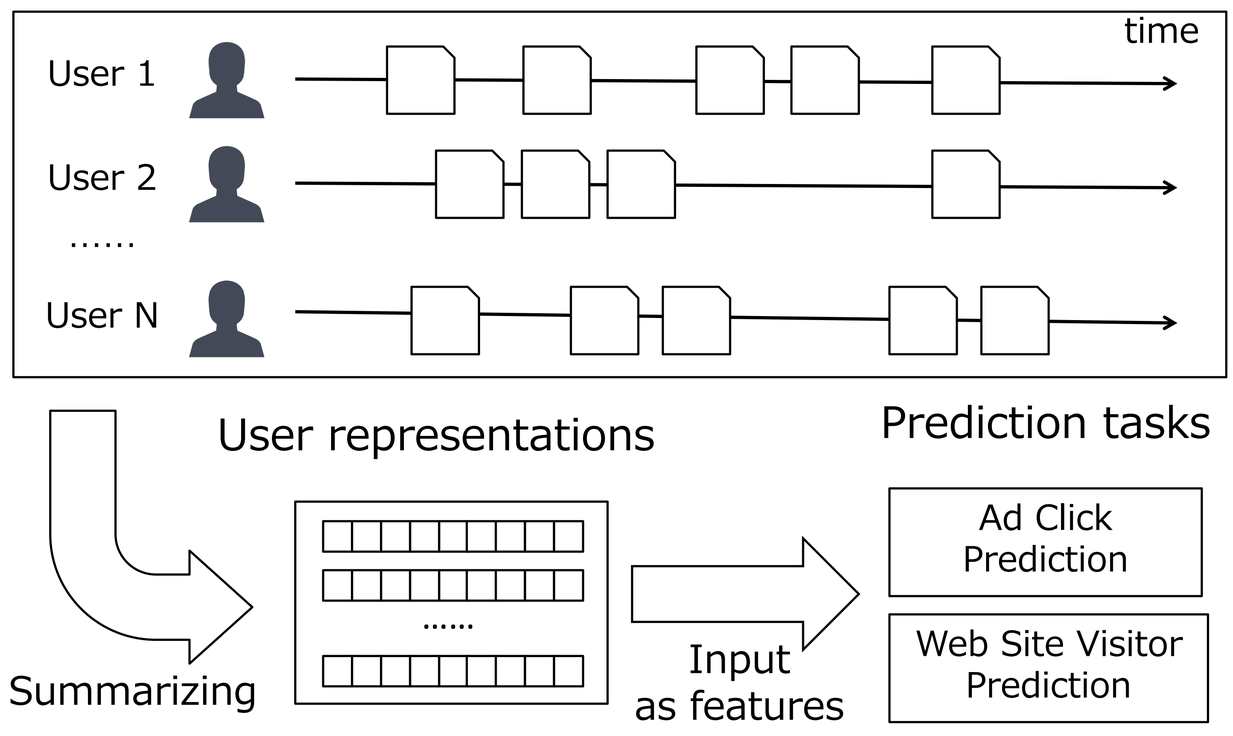
Yukihiro Tagami, Hayato Kobayashi, Shingo Ono, Akira Tajima
{yutagami, hakobaya, shiono, atajima}@yahoo-corp.jp

Background

- Large-scale Web sites that provide various Web services deal with a lot of user-related prediction tasks. However,
  - Some of these tasks have small-scale training data
  - User activities on the Web site are ID-based (such as URLs), and too sparse to be used as features for such cases

- We propose an approach to obtain low-dimensional user vectors from sequences of user activities using recent representation learning approaches in NLP field
  - Paragraph Vector [3] along with Skip-gram model [4]
  - Considering users and activities as paragraphs and words
  - In this poster, we focus on Web page visits as user activities and represent it as the URL of the Web page

◆ Overview of our approach



Paragraph Vector

◆ PV-DM (Distributed Memory model of Paragraph Vector)

- The objective of the vector models for a sequence of *i*-th user activities is to maximize the sum of log probabilities

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

- The PV-DM defines the probability using the softmax function

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

$\boldsymbol{w}_{a_{i,t}}$: "output" vector corresponding to activity $a_{i,t}$

$\boldsymbol{v}_{a_{i,t}}$: "input" vector corresponding to activity $a_{i,t}$

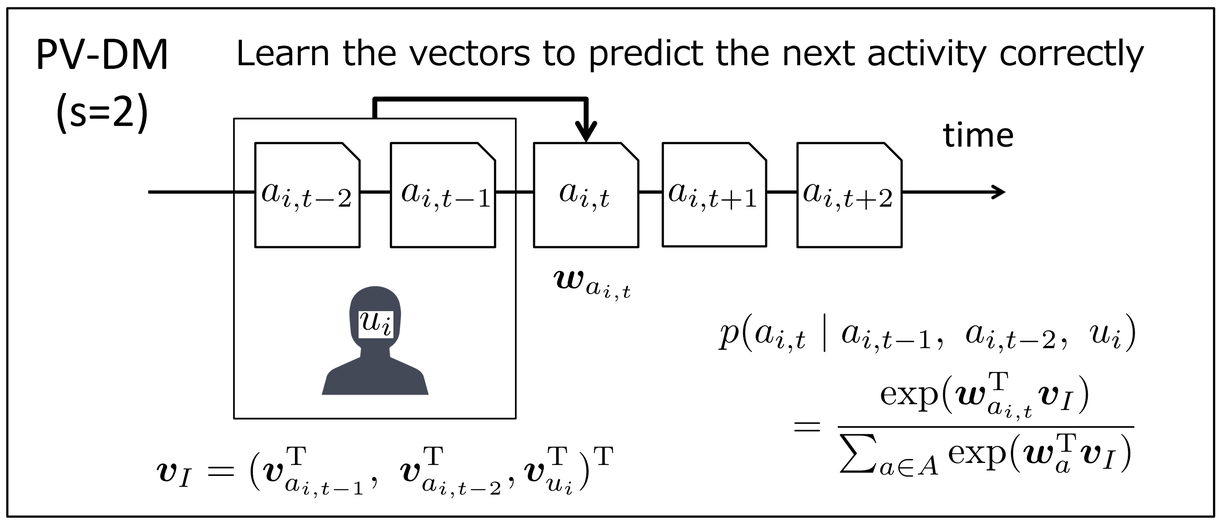
$\boldsymbol{v}_{u_i}$: "input" vector corresponding to user u_i

$\boldsymbol{v}_I = [\boldsymbol{v}_{a_{i,t-1}}^{\mathrm{T}}, \ldots, \boldsymbol{v}_{a_{i,t-s}}^{\mathrm{T}}, \boldsymbol{v}_{u_i}^{\mathrm{T}}]^{\mathrm{T}}$: concatenated input vector

- We employ negative sampling approach for fast training

$$\log \sigma(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I) + k \cdot \mathbb{E}_{a_n \sim p_n(a)} \left[\log \sigma(-\boldsymbol{w}_{a_n}^{\mathrm{T}} \boldsymbol{v}_I)\right] \qquad \sigma(z) = \frac{1}{1+\exp(-z)}$$

- Settings of learning log-bilinear models
  - About one billion page visits (about 3.52 million unique URLs)
  - The size of input vectors: 400, the size of context window: 5, the number of sampled negative instances: 5



Experiments

◆ Data sets (prediction tasks)

- AdClicker
  - Consisting of the users who clicked contextual ads that are included in the five selected ad campaigns (Ac1–Ac5)
- SiteVisitor
  - Consisting of the users who visited Web sites of five selected advertisers (Sv1–Sv5)

	#Training	#Validation	#Testing	#Feature
AdClicker	51,576	10,000	10,000	786,467
SiteVisitor	1,862,693	20,000	20,000	17,574,741

- We transformed the multi-label problems into a set of binary classification problems and trained logistic regression classifiers
  - The evaluation measure is Area Under ROC curve (AUC)

◆ Methods to extract user representations

- Bag of URLs (high-dimensional vectors)
  - Bin: whether the user visited each Web page or not (1/0)
  - Freq: frequencies of the user's each Web page visits
- Log-bilinear models (low-dimensional vectors)
  - Skip-gram: simple averaging of activity vectors
  - PV-DM: proposed method using $\boldsymbol{v}_{u_i}$
  - PV-DM+Skip-gram: concatenated vectors of above two

◆ Results

- PV-DM achieved better results than Skip-gram in SiteVisitor whereas the opposite trend is shown in AdClicker
- The combination method PV-DM+Skip-gram performed better than individual methods

	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

WWW 2015, Florence, Italy