Paragraph Vectorを用いたウェブ上のユーザー行動のモデリング

田頭 幸浩　小林 隼人　小野 真吾　田島 玲
ヤフー株式会社
{yutagami, hakobaya, shiono, atajima}@yahoo-corp.jp

1 導入

近年，単語の分散表現を教師無し学習によって得る手法が注目を浴びている [4, 5]．この手法で得られた単語ベクトルは統語的や意味的な関係を捉えた表現になっており，類推タスクなどでの有効性が示されている [4, 5]．また，パラグラフや文書などの単語より大きな粒度での分散表現を得る手法として，Paragraph Vector [3] が提案されている．この手法で得られたベクトル表現も同様に，各種予測タスクの有効な素性になるとの報告がなされている [3]．

本稿では，Paragraph Vector を用いてウェブ上のユーザーの行動列を集約するアプローチを提案する．このアプローチでは，ユーザーをパラグラフもしくは文書，ユーザー行動を単語と見なし，ユーザーの行動列に対してこの自然言語処理の手法を適用する．学習で得られたユーザーを表現する低次元の素性ベクトルは，ニュース記事のレコメンデーションや広告のクリック率の予測などのユーザーに関連した各種予測タスクで共通して用いることができる．ユーザーのベクトル表現は行動ログから教師無し学習で得ることができるため，ウェブサイトやスマートフォンアプリ全体ではユーザーの行動ログが潤沢に得られる一方で，個々の予測タスクの学習データが少ない場合に，このベクトル表現は有効な素性になると期待される．このアプローチを Yahoo! JAPAN のログをもとに作成した 2 種類のデータセットを用いて評価を行い，その有効性を確認する．

2 提案手法

この章では，対象とする問題設定にしたがって Paragraph Vector [3] について説明する．先述した通り，ユーザーをパラグラフもしくは文書，ユーザー行動を単語と見なし，このベクトルモデルをユーザーの行動列に対して適用する．ユーザーの行動の例として，ウェブページ訪問や検索クエリの入力，広告クリックを挙げることができる．i 番目のユーザー u_i のウェブ上での行動列を $(a_{i,1}, a_{i,2}, \ldots, a_{i,T_i})$ と表す．なお，$a_{i,j}$ はユーザー u_i の j 番目の行動，T_i は行動列の長さを表す．この行動列に対して，Paragraph Vector のモデルは以下の対数確率の平均値を目的関数とし，最大化を行う．

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

なお，s はコンテキストウィンドウのサイズである．このマルチクラス問題の確率は一般的に，softmax 関数を用いて表現される．Paragraph Vector のモデルの一つである PV-DM（Distributed Memory Model of Paragraph Vectors）は，この確率を以下のように log-bilinear モデルとして定義する．

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

ここで，$\boldsymbol{w}_{a_{i,t}}$ は行動 $a_{i,t}$ に対応する出力ベクトルを，$\boldsymbol{v}_I$ は対象の時刻以前の行動に対応する入力ベクトル $\boldsymbol{v}_{a_{i,t-1}}, \ldots, \boldsymbol{v}_{a_{i,t-s}}$ とユーザーに対応する入力ベクトル $\boldsymbol{v}_{u_i}$ を連結したベクトルを表し，$\boldsymbol{v}_I = [\boldsymbol{v}_{a_{i,t-1}}^{\mathrm{T}}, \ldots, \boldsymbol{v}_{a_{i,t-s}}^{\mathrm{T}}, \boldsymbol{v}_{u_i}^{\mathrm{T}}]^{\mathrm{T}}$ である．A はユーザーが取りうる行動の集合を表す．$j \le 0$ となる $\boldsymbol{v}_{a_{i,j}}$ は特別なベクトル $\boldsymbol{v}_{NULL}$ で置き換える．

行動に対応する入力ベクトルのサイズ $|\boldsymbol{v}_{a_{i,j}}|$ を v_a，ユーザーに対応する入力ベクトルのサイズ $|\boldsymbol{v}_{u_i}|$ を v_u と定義すると，連結した入力ベクトル $\boldsymbol{v}_I$ と出力ベクトル $\boldsymbol{w}_{a_{i,j}}$ のサイズはともに $s \times v_a + v_u$ である．学習で得られたユーザーに対応する入力ベクトル $\boldsymbol{v}_{u_i}$ を，ニュース記事のレコメンデーションや広告クリック予測などのさまざまな予測タスクの素性として用いる．

一般的に行動の種類数 $|A|$ は多く，式 (1) やその一

階導関数の計算コストは高いため，そのまま計算することは実用的ではない．そのため，Le と Mikolov [3] は単語の頻度を考慮した階層 softmax を用いることで学習の高速化を行った．ここでは階層 softmax の代わりに，negative sampling [5] のアプローチをとる．このアプローチでは，式（1）を代入した目的関数 $\log p(a_{i,t} \mid a_{i,t-1},\ \ldots,\ a_{i,t-s},\ u_i)$ を以下の式で近似する．

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

なお，$\sigma(z) = 1/(1+\exp(-z))$ は sigmoid 関数であり，k は negative sample の数，$p_n(a)$ は negative sample を生成する分布である．Mikolov ら [5] に従い，$p_n(a)$ として単語の頻度分布 $U(a)$ を 3/4 乗した分布を用いる．このモデルを AdaGrad [2] を用いた非同期 SGD（Asynchronous Stochastic Gradient Descent）[6] で学習する．新しいユーザーに対する推論時には，行動に対応する入力と出力のベクトル $\boldsymbol{v}_a$ と $\boldsymbol{w}_a$ を固定して，ユーザーに対応するベクトル $\boldsymbol{v}_u$ のみを同様に学習する．

上記のアプローチに加えて，Paragraph Vector のモデルの一つである，PV-DBoW（Distributed Bag of Words version of Paragraph Vector）で得られたベクトルを予測タスクの素性として用いるアプローチも実験で用いる．

3 実験

この章では，提案手法の評価実験を行う．まず，データセットと実験設定，比較手法について述べ，その後，実験結果を示し考察を行う．

3.1 データセット

提案手法を評価するため，教師あり学習のデータセットとして *AdClicker* データセットと *SiteVisitor* データセットの二つを用意した．*AdClicker* データセットは，五つの広告キャンペーンに含まれるコンテキスト広告をクリックしたユーザーの集合からなる．同様に，*SiteVisitor* データセットは，五つの広告主のウェブサイトを訪れたユーザーの集合から構成される．

これら二つのデータセットは，前日のウェブページ訪問をもとに，その翌日のユーザーの行動を予測する問題として作成した．学習データとバリデーションデータは 2014 年 7 月 22 日と 23 日のログから作成した．22 日のウェブページ訪問を素性として用い，23 日の広告クリックや広告主のサイト訪問をラベルとした．同様に，テストデータは 2014 年 7 月 23 日と 24 日のログから作成した．上記の素性は Yahoo! JAPAN のウェブサービスのログから抽出したものであるため，ウェブ上でのユーザーの行動のほんの一部分であることに注意が必要である．また，*SiteVisitor* データセットのラベルである，広告主のサイト訪問は含まれない．

一人のユーザーが複数の広告をクリックしたり，さまざまな広告主のサイトを訪問することがあり得るため，これら二つのデータセットはマルチラベルのデータセットである．実験では，マルチラベル問題を二値分類問題の集合に変換し，それぞれの手法で抽出された素性を用いてロジスティック回帰のモデルを学習し，AUC（Area Under ROC Curve）を指標として評価した．二つのデータセットの統計量を表 1 にまとめた．

3.2 実験設定と比較手法

2014 年 7 月 22 日のログの一部を用いて PV-DM モデルを学習し，得られたユーザーベクトルを教師あり学習タスクの素性として用いた．上記の抽出データの中で出現回数が 5 回を下回るウェブページ訪問を削除した．二つの連続するページ訪問の時刻の差が 30 分以上の場合は，セッションの区切りを示す特殊なシンボルを挿入した．結果として，得られたデータ中の URL のユニーク数はおよそ 352 万，ページ訪問の総数は約 10 億となった．PV-DM の学習時の設定は，入力ベクトルのサイズを $v_a = v_u = 400$，コンテキストウィンドウのサイズを $s = 5$，ネガティブサンプルの数を $k = 5$，エポック数（SGD におけるデータの周回数）を 5 とした．

AdClicker データセットと *SiteVisitor* データセットの学習データとバリデーションデータは，教師無し学習のデータと同じく 2014 年 7 月 22 日のログから作成したため，それらのデータに含まれるユーザーのベクトルは PV-DM モデルの学習時に得られるが，ここではテストデータに含まれる未知のユーザーと同様に推論のステップを経てユーザーベクトルを獲得し，素性ベクトルとした．

実験では Paragraph Vector を用いたアプローチといくつかのベースラインを比較した．*Bin* と *Freq* は URL をそのまま素性として用いた手法である．*Freq* はユーザーのページ訪問の頻度を考慮するのに対し，*Bin* はウェブページに訪れたか否かのみを用いた．言い換えると，*Bin* はバイナリ素性，*Freq* は Term Frequency を素性とした場合に対応する．*Skip-gram* は Skip-gram

表 1: *AdClicker* と *SiteVisitor* データセットの統計値．#Features は各データセットに含まれるユニークな URL 数．

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

モデル [5] の学習で得られたベクトルを用いた手法である．このアプローチでは，Djuric ら [1] が行ったように，あるユーザーをその行動列に含まれる行動ベクトルを単純に平均したベクトルで表現する．PV-DM と PV-DBoW を用いた提案手法をそれぞれ *PV-DM* と *PV-DBoW* で表す．Skip-gram と PV-DBoW の学習データや設定は，PV-DM のものと同じである．上記に加えて，Le と Mikolov の薦め [3] にしたがい，PV-DM と PV-DBoW モデルで学習したベクトルを連結して素性として用いた手法の評価も行った．この手法を *PV-DM+PV-DBoW* と記す．非同期 SGD の確率的振る舞いと初期値の依存性のため，*PV-DM* と PV-DBoW，*Skip-gram*，*PV-DM+PV-DBoW* の結果は，5 回の実行結果の平均値を用いた．

3.3 実験結果

実験結果を表 2 にまとめた．**太字**の項目は，各手法のうち最大の結果を示している．また，下線 の項目は，*PV-DM* と *PV-DBoW*，*Skip-gram* の三つのうち，良い結果を示している．

PV-DM+PV-DBoW は全 10 タスク中 9 の場合において一番良い結果であった．*SiteVisitor* データセットでは *PV-DM* が *Skip-gram* よりも良い結果を示したが，*AdClicker* では逆の傾向が見られた．*AdClicker* データセットに含まれるコンテキスト広告は，ユーザーの情報だけでなくウェブページのコンテンツも考慮して表示されるため，これらの予測タスクにおいては Skip-gram モデルを用いてウェブページの表現を学習することが有効であったと考えられる．一方で，より複雑なユーザーの興味を反映している *SiteVisitor* データセットにおいては，訪問系列の順番や全体を考慮した PV-DM モデルを用いたアプローチが有効であったと考えられる．Paragraph Vector のモデルである *PV-DM* と *PV-DBoW* を比較すると，タスクにより有効な手法は異なっていた．一方で，モデルで得られたベクトルを連結すると，個々のモデルで得られたベクトルのみを用いた場合と比較して，安定して良い結果を示したため，それら二つのモデルはユーザー行動の異なる側面を学習したのだと考えられる．素性として URL をそのまま用いる *Bin* と *Freq* は，ほとんどの場合において劣った結果となった．

4 まとめと今後の課題

本稿では，ユーザーの行動履歴から教師無し学習によってユーザー表現を得ることを目的とし，Paragraph Vector [3] を用いてユーザーの行動列を集約するアプローチを提案した．Yahoo! JAPAN のログをもとに作成した 2 種類の予測タスクのデータセットを用いて実験を行い，このアプローチを評価した．ユーザーが訪れた広告主のサイトを予測するタスクでは，PV-DM モデルを用いた手法が Skip-gram モデルを用いた手法よりも良い結果を示したが，ユーザーがクリックした広告を予測するタスクでは逆の傾向が見られた．二つの Paragraph Vector のモデルで得られたベクトルを連結すると，個々のモデルで得られたベクトルのみを用いた場合と比較して安定して良い結果を示したため，それら二つのモデルはユーザー行動の異なる側面を学習したのだと考えられる．

本稿は，自然言語処理で提案された分散表現を用いた手法を，他の分野のタスクにも適用可能であることを示した．一方で，今回対象としたユーザーのウェブページ訪問の系列はウェブページのリンク構造に大きく依存しており，自然言語の文と比較して系列中のパターンの自由度は制限されていると考えられる．その違いが分散表現を獲得するためのモデルや学習にどのような影響を与えているかについての分析は，今後の課題である．

参考文献

[1] N. Djuric, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hidden conditional random fields with distributed user embeddings for ad targeting. In *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2014.

[2] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochas-

表 2: 実験結果．値は評価指標である AUC を表す．

	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-DBoW	0.9891	0.8329	0.6555	0.7300	0.7578	0.7976	0.8303	0.7192	0.7927	0.7489
PV-DM+PV-DBoW	0.9901	**0.8370**	**0.6614**	**0.7461**	**0.7736**	**0.8101**	**0.8386**	**0.7280**	**0.8051**	**0.7546**

tic optimization. *The Journal of Machine Learning Research*, 12, 2011.

[3] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *The 31st International Conference on Machine Learning*, 2014.

[4] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, 2014.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, 2013.

[6] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*, 2011.