Event2Vec: Learning Event Representations Using Spatial-Temporal Information for Recommendation

Yan Wang and Jie Tang

Department of Computer Science and Technology, Tsinghua University, China
yan-w16@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn

Abstract. Event-based social networks (EBSN), such as meetup.com and plancast.com, have witnessed increased popularity and rapid growth in recent years. In EBSN, a user can choose to join any events such as a conference, house party, or drinking event. In this paper, we present a novel model—Event2Vec, which explores how representation learning for events incorporating spatial-temporal information can help event recommendation in EBSN. The spatial-temporal information represents the physical location and the time where and when an event will take place. It typically has been modeled as a bias in conventional recommendation models. However, such an approach ignores the rich semantics associated with the spatial-temporal information. In Event2Vec, the spatial-temporal influences are naturally incorporated into the learning of latent representations for events, so that Event2Vec predicts user's preference on events more accurately. We evaluate the effectiveness of the proposed model on three real datasets; our experiments show that with a proper modeling of the spatial-temporal information, we can significantly improve event recommendation performance.

1 Introduction

Event-based social network (EBSN) is a new type of social network that has experienced increasing popularity and rapid growth. For instance, Meetup[1], one of the largest online social networks for facilitating offline group meetings, has attracted 30 million registered users who have created nearly 270,000 Meetup groups. Douban[2], a Chinese social networking service, has more than 200 million registered users and has hosted about 590,000 offline groups. These EBSN websites allow members to find and join groups unified by a common interest, such as politics, books, games, movies, health, careers or hobbies, and schedule a time to meet up together offline, which results in very interesting user behavior data combining both online and offline social interactions [9]. One challenging issue on these EBSN websites is how to keep users actively joining new events. Recommendation plays a critical role [11].

In contrast to conventional online social networks that mainly contain user's online interactions, users in EBSN can choose to join the event according to their interest in the event (based on the event content) and their availability (based on the event location and availability at the schedule time). Therefore, user's mobile behaviors presented

[1] https://meetup.com
[2] https://douban.com

in EBSN are explored typically in several important aspects, including event content, spatial influence [8] and temporal effect [5].

Many recent studies have exploited different factors to improve recommendation effectiveness. For instance, some efforts have been made to explicitly model the spatial information as in [15][20]. Some others exploit temporal cyclic effect to provide spatial or/and temporal novel recommendation like [18]. However, they lack an integrated analysis of the joint effect of all factors in a unified effective way and no previous work has explicitly modeled user's preference on both spatial and temporal factors to improve the recommendation performance.

In this work, we stand on the recent advances in embedding learning techniques and propose an embedding method—Event2Vec to encode events in a low-dimension latent space which integrates the spatial and temporal influence. In specific, we learn representations for three factors—the event, the location and the time simultaneously using the event sequential data attended by users. We propose to use multitask learning settings to model and predict user's preference on three factors naturally. The technique of shared embeddings are utilized in our proposed model to improve the efficiency.

In addition, our approach leverages the interactive influence between spatial and temporal factors presented in user's behaviors by modeling the combination of spatial-temporal information. In specific, events held at the same location could have very different topics at different time periods, thus attract varying groups of user. For instance, an urban park usually holds events like "picnic" in the afternoon while holds events like "jogging" at night. In the course of this paper, we will present how our embedding model exploits such joint and interactive influences of spatial and temporal factors in a natural way.

Finally, we propose a recommendation algorithm based on a similarity metric in the latent embedding space which is proved to be effective in our experiments. Compared with state-of-the-art recommendation frameworks, we can achieve a significant improvement.

2 Problem Definition

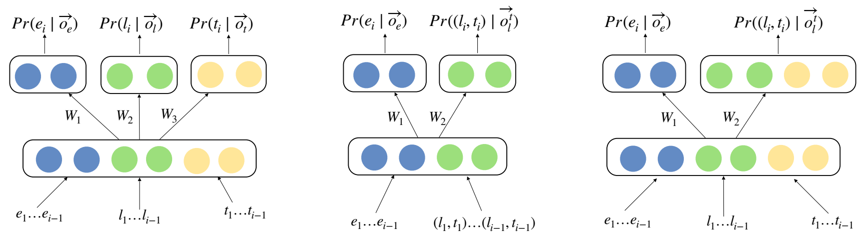
In this section, we will first clarify some terminology used in this paper, and then explicitly present our problem.

User behaviors are formulated as a set of four tuple $\{(u, e, l, \tau) : u \in U, e \in E, l \in L, t \in T\}$, where each means user u attended event e at location l, at time slot t. U is a set of users and E is a set of events, L [3] is a set of locations and T is a set of time slots discretized from continuous timestamps. We use the notation $|\cdot|$ to denote the cardinality of a set — for example, $|L|$ indicates of the number of locations in set L.

For each user u, we create a user profile $D_u = \{(e_i, l_i, t_i), i = 1 \ldots n_u\}$, which is a sequence of events user u attended in chronological order.

Input: The input of our problem is an event-based social network $G = (U, E, L, T)$, and a set of user profiles $D = \{D_u : u \in U\}$.

[3] The location l can be represented as a pair (longitude, latitude) or a specific address (e.g., "Wine Bar at MIST").



(a) Event2Vec-1 (b) Event2Vec-2 (c) Event2Vec-3

Fig. 1. Architectures of the three Event2Vec models

Goal: Given a querying user u, our goal is to recommend upcoming events based on historical preferences of the user .

3 The Proposed Approach

In this section, we present the details of the proposed model—Event2Vec.

To incorporate different types of information, we learn latent representations for each event, location and time. Then, we model and predict user's preferences on the three factors explicitly to improve the recommendation accuracy.

In specific, the three factors are related to each other: for instance inferring user's preference on the location helps the inference of user's preference on the time. Predicting one helps in predicting the other one, and three factors altogether decides user's tendencies and behaviors. Therefore, we propose to take the perspective of multitask learning settings to naturally leverage the useful information contained in user's preferences on different factors which are related to each other. We set up three single tasks for predicting user's preference on the event, the location and the time respectively. We propose to use shared parameters (i.e., shared embeddings) in all three different tasks to learn latent representations which integrate different points of view. Shared embeddings are also important for the efficiency and generalization of low-dimensional representation learning in our proposed model.

We derive three different model architectures each with different target variables to implement the proposed model.

In the first model (Event2Vec-1, Fig. 1(a)), we learn the embeddings for each event, location and time by learning to predict the next event user would attend, and the associated location and time simultaneously.

In the second model (Event2Vec-2, Fig. 1(b)), we learn the embeddings for each event and spatial-temporal pair (i.e., (l, t)) to further capture the interactive influences between spatial and temporal factors.

In the third model (Event2Vec-3, Fig. 1(c)), we propose a compromise between Event2Vec-1 and Event2Vec-2. We reserve distinct embeddings for each location and time but predict the spatial-temporal pair as a combination.

In the remainder of this section, we will describe the three models in more detail.

3.1 Our Models

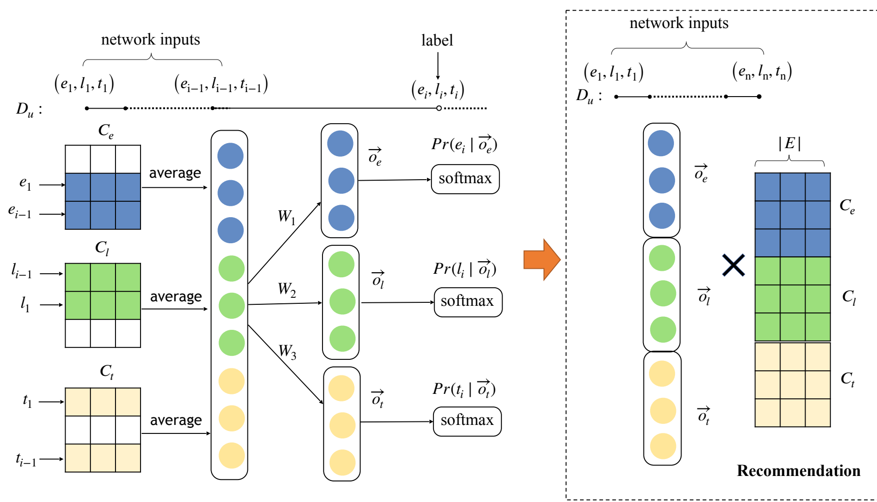


Fig. 2. The joint model with three target variables in which three similar networks are trained per each target variable. At serving, a nearest neighbor lookup is performed to generate a set of event recommendations.

Event2Vec-1 . Event2Vec-1 learns low dimensional embeddings for each event, location and time in a fixed vocabulary and feeds these embeddings into a feedforward neural network. The purpose of the neural network is to predict user's next behavior including the event to attend, the location and the time to go, using his/her historical behaviors. A user's history is represented by a variable-length sequence of sparse event, location and time IDs which are mapped to dense vector representations via the embeddings. However the network requires fixed-sized dense inputs. We find averaging the embeddings performed best among several strategies (sum, component-wise max, etc.).

More formally, we describe the proposed model starting with a single network of predicting the next event. Given a user profile $D_u = \{(e_j, l_j, t_j), j = 1 \dots n_u\}$, to predict users' preferences on the event, the input is a sequence of $\{(e_1, l_1, t_1) \dots (e_i, l_i, t_i)\}$ and the target is e_{i+1}, where i ranges from 1 to $n-1$. we feed the sequence into the neural network, which are represented by three one-hot vectors for the event, location and time respectively. The entry is set to one if it exists in the sequence, zero otherwise. In the embedding layer, we look up the embeddings from three embedding matrices, i.e., $C_e \in \mathbb{R}^{|E| \times d_e}$, $C_l \in \mathbb{R}^{|L| \times d_l}$ and $C_t \in \mathbb{R}^{|T| \times d_t}$, where d_e, d_l and d_t are the dimensions of the event, location and the time representations. By averaging, three fixed-sized vectors $\overrightarrow{e_{avg}}$, $\overrightarrow{l_{avg}}$ and $\overrightarrow{t_{avg}}$ are obtained. Then they are concatenated into a flat vector $\overrightarrow{v_{in}}$ which is fed as the input of the following fully-connected layers, with

$|\overrightarrow{v_{in}}| = d_e + d_l + d_t$. We use one fully-connected layer parameterized by W_1 in our model.

The output of the fully-connected layer, denoted as $\overrightarrow{o_e} \in \mathbb{R}^{d_e}$, encodes the user's historical behaviors and thus can be used to predict the upcoming events user will attend. Let e_i denotes the target event, given the encoded historical behaviors $\overrightarrow{o_e}$, our model formulates the conditional probability $Pr(e_i \mid \overrightarrow{o_e})$ using a softmax function in Eq. 1.

$$Pr(e_i \mid \overrightarrow{o_e}) = \frac{\exp(\overrightarrow{e_i}^T \cdot \overrightarrow{o_e})}{\sum_{e' \in E} \exp(\overrightarrow{e'}^T \cdot \overrightarrow{o_e})} \tag{1}$$

where $\overrightarrow{e_i}$ and $\overrightarrow{e'}$ are row vectors of C_e. In order to make the model efficient for learning, the techniques of hierarchical softmax and negative sampling are used as proposed in Skip-Gram [6]. Similar to the single network of predicting the event, the other two neural networks with target variables of the location and time are built and output the probabilities of $Pr(l_i \mid \overrightarrow{o_l})$ and $Pr(t_i \mid \overrightarrow{o_t})$. Therefore, the objective of Event2Vec-1 is to minimize three cross entropy losses simultaneously. Fig. 1(a) illustrates the architecture of Event2Vec-1 model.

At serving time we need to recommend top k events to the user. Our recommendation algorithm is based on the user-event cosine similarity in the embedding space. Since both spatial and temporal factors play important roles in event recommendation, so we utilize all output vectors of the neural networks to make recommendations.

In specific, we feed all user's historical behaviors into the neural networks and obtain the predicted vectors $\overrightarrow{o_e}$, $\overrightarrow{o_l}$ and $\overrightarrow{o_t}$ by forward propagation. We build user's preference $\overrightarrow{v_u}$ by concatenating them all together, i.e., $\overrightarrow{v_u} = \overrightarrow{o_e} \| \overrightarrow{o_l} \| \overrightarrow{o_t}$, where $\|$ is the concatenation operation. For each candidate event e_i associated with location l_i and time t_i, we get its final representation as $\overrightarrow{v_{e_i}} = \overrightarrow{e_i} \| \overrightarrow{l_i} \| \overrightarrow{t_i}$, where the embeddings are looked up in the embedding matrices—C_e, C_l and C_t.

Given a user u, for each event e_i which has not been attended by u, we compute its ranking score using Eq. 2, and select top k events with highest scores to recommend to the user.

$$S(u, e_i) = \overrightarrow{v_u}^T \cdot \overrightarrow{v_{e_i}} \tag{2}$$

Fig. 2 demonstrates our proposed joint model with three target variables in which three similar networks are trained per each target variable. The trainable parameters include three embedding matrices, C_e, C_l and C_t, and the weight matrices of the fully connected layer, W_1, W_2 and W_3. Please note that parameters of the embedding matrices are shared and trainable in all three neural networks, while parameters of weight matrices are only updated through the associated neural network.

Event2Vec-2 . The combination of the location and the time contain richer semantic information, however Event2Vec-1 doesn't consider such interactive influence between the spatial and temporal factors. A location usually holds different semantics at different time, and these semantics should have discriminative vectors. Therefore, Event2Vec-2 learns embeddings for each spatial-temporal pair. The spatial-temporal embedding matrice is denoted as $C_l^t \in \mathbb{R}^{|L \times T| \times d_l^t}$, where d_l^t means the dimension of spatial-temporal representation and $L \times T$ means the Cartesian product of L and T.

The architecture of Event2Vec-2 is illustrated in Fig. 1(b). There are two neural networks predicting the next event and the next spatial-temporal pair respectively. When making recommendations, the user preference is represented as $\overrightarrow{v_u} = \overrightarrow{o_e} \| \overrightarrow{o_l^t}$; and the candidate event e_j is represented as $\overrightarrow{v_{e_i}} = \overrightarrow{e_i} \| \overrightarrow{l_i^t}$, where $\overrightarrow{e_i}$ and $\overrightarrow{l_i^t}$ are row vectors of C_e and C_l^t.

Event2Vec-3 . Since Event2Vec-2 divides the occurrences of each location into multiple time slots, the learning of embeddings suffer from the sparsity issue. In an attempt to alleviate the problem, we propose a new model—Event2Vec-3 to provide a trade-off between the discrimination and sparsity.

Event2Vec-3 reserves distinct embeddings for each event, location and time. However slightly different from Event2Vec-1, the location and the time are predicted as a combination. Each spatial-temporal pair (l, t) is represented by concatenating their distinct vectors $\overrightarrow{l}$ and $\overrightarrow{t}$ into a flat vector $(\overrightarrow{l} \| \overrightarrow{t}) \in \mathbb{R}^{d_l+d_t}$. The corresponding output $\overrightarrow{o_l^t}$ has the same length of $d_l + d_t$. The outputs of two neural networks are $Pr(e_i \mid \overrightarrow{o_e})$ and $Pr((l_i, t_i) \mid \overrightarrow{o_l^t})$ as shown in Fig. 1(c), where $Pr((l_i, t_i) \mid \overrightarrow{o_l^t})$ is calculated as in Eq. 3

$$Pr((l_i, t_i) \mid \overrightarrow{o_l^t}) = \frac{\exp((\overrightarrow{l_i} || \overrightarrow{t_i})^T \cdot \overrightarrow{o_l^t})}{\sum_{(l', t') \in L \times T} \exp((\overrightarrow{l'} || \overrightarrow{t'})^T \cdot \overrightarrow{o_l^t})} \tag{3}$$

It's worthy of noting that the embeddings of each location and time are shared among all spatial-temporal pairs (l, t).

4 Experiments

In this section, we evaluate the proposed model for the task of event recommendations. We first examine the performance of Event2Vec models compared with related models in Section 4.2. Then we examine the importance of spatial-temporal factors in Section 4.3; and finally different temporal patterns are compared and discussed in Section 4.4.

4.1 Experimental Setup

Datasets. We use three datasets in real-world domains, two from Douban and one from Meetup, for our experiments.

- **Meetup.** We collected the first dataset *Meetup* by crawling real events hosted in New York from meetup.com in 2016. For each event, we retrieved its geographic location, start time, and a list of users who attended. To reduce noise, we selected events that are attended by at least 20 users, and users who have attended at least 20 events. In the end, the *Meetup* dataset contains 4722 users and 5064 events.

– **Douban [19].** We collected two datasets *Douban-bej* and *Douban-sha* by crawling events hosted in 2012 from douban.com located at Beijing and Shanghai respectively. For each event, we also retrieved its geographic location, start time, and a list of registered users who attended. Then we removed users who attended fewer than 20 events, and events attended by fewer than 20 users. We have 222795 attendances by 6513 users, 5326 events in the *Douban-bej* dataset; 6964 users, 4189 events and 241093 attendances in the *Douban-sha* dataset.

Data Preprocessing. To normalize the locations of events, we split the city into even grid cells according to coordinates, and each resultant location (gird) spans 0.13km. The numbers of locations in the *Meetup*, *Douban-bej* and *Douban-sha* dataset are 1569, 813 and 626 respectively.

To capture the temporal characteristics in user's behaviors, we design a time discretizing scheme to smoothly map a continuous timestamp to a time slot. The preference variance exists in three time scales generallly: hours of a day, different days in a week (or a month), and different months in a year, which is observed in (Gao et al. 2013) but not modeled. By experiments, we propose to divide the continuous time space into time slots using a weekday-hour pattern, such as "4 (day of the week), 1:00-2:00 (hour of the day)". Therefore, we can get at most 7*24 discretized time slots on all three datasets. Other temporal patterns are compared and discussed in Section 4.4.

Comparison Methods. We compare our model with the following methods representing the state-of-the-art event-based recommendation techniques.

– **SVDFeature.** SVDFeature [3] is a machine learning toolkit designed to solve the feature-based matrix factorization. To compare with our model fairly, we implement it by incorporating more side information including the location and the time.
– **IRenMF.** IRenMF [10] is based on Weighted Matrix Factorization (WMF). IRenMF considers the influence of neighboring locations while modeling user's preferences.
– **Rank-GeoFM.** Rank-GeoFM [7] is a ranking based factorization method, which includes spatial influence in a latent model.
– **Event2Vec.** Our proposed methods for event recommendation, which incorporate spatial-temporal information using the embedding learning methods.

In summary, SVDFeature models the spatial-temporal information as simple bias, while both IRenMF and Rank-GeoFM model geographic influences as latent vectors using Matrix Factorization techniques.

For each individual user in the dataset, we sort his behaviors in time order and then mark off the last 10% events he attended for testing, while use the previous 90% historical events for training. In the experiments, we use a validation set to find the optimal hyper-parameters, and finally set d_e, d_l and d_t to 200, (we use the same dimension for simplicity, but they are not necessarily equal in practice). For implementation, we develope the model based on Tensorflow [1]. We use stochastic gradient descent (SGD) for optimization, and gradients are calculated using the back-propagation algorithm. We run each recommendation method for 5 times and report the average performances in Table 1.

Evaluation Metrics. We compare the performances through precision, recall, and f1-score as they are generally used in recommendation systems. We denote these metrics at top-k recommendation as p@k, r@k, f1@k respectively. Formally, if we define E_u^R as recommended events sorted by score in descending order and E_u^T as the true events attended by user u,

$$\begin{aligned} \mathrm{p}@k &= \frac{1}{|U|} \sum_{u \in U} \frac{|E_u^T \cap E_u^R[:k]|}{k} \\ \mathrm{r}@k &= \frac{1}{|U|} \sum_{u \in U} \frac{|E_u^T \cap E_u^R[:k]|}{|E_u^T|} \\ \mathrm{f1}@k &= \frac{2 \cdot \mathrm{p}@k \cdot \mathrm{r}@k}{\mathrm{p}@k + \mathrm{r}@k} \end{aligned} \tag{4}$$

4.2 Results

Table 1 shows the experimental results. We find Event2Vec models outperform other baselines significantly on all metrics, among which Event2Vec-2 achieves the best performance. The standard deviation of the performance from each method is less than 4×10^{-4}, confirming the reliability of our comparison results.

Baselines vs. Our Models. Several observations are made by comparing baselines and our models from the results. (1) Rank-GeoFM and IRenMF achieve a higher recommendation accuracy than SVDFeature on all metrics of performance, showing the benefits brought by factorizing the spatial-temporal influences into latent vectors instead of scalar bias used by SVDFeature. (2) Event2Vec models outperform other competitor methods by 4%-9% in terms of $p@10$ on three datasets. It shows the advantages of the proposed multitask learning framework and shared embeddings in modeling different related factors. Moreover, the proposed Event2Vec models explicitly predict user's preferences on three factors using the historical data. Therefore, we can see a significant improvement over other baseline methods in Table 1.

Event2Vecs. The performance of three Event2Vec models are very different and reflect their characteristics. (1) Event2Vec-2 achieves the best performance. Event2Vec-2 outperforms Event2Vec-1 by 0.9%-2.7% in terms of f1-score. The most possible reason is, Event2Vec-2 discriminates different location-time combinations and learn distinct representations for each of them to capture more accurate semantics. For example, the representation of "cafe-morning" learned by Event2Vec-2 could encode concrete and discriminative semantics probably like "breakfast", while in Event2Vec-1 it's represented by concatenating the vectors of "cafe" and "morning" which may introduce the noises. From the results, we can conclude that in Event2Vec-2, the effectiveness of modeling interactive influence between the spatial and temporal factors is more significant than the issue caused by sparsity, thus Event2Vec-2 achieves the best performance. (2) The performance of Event2Vec-3 drops behind the other two Event2Vec methods, this is probably because during the back propagation, the updates on embeddings of the location and the time will influence each other, for that the boundary of the embeddings are blurred because of concatenating operation. Therefore it makes the representation

Table 1. Performance Comparison

Dataset	Meetup						
Metric	p@1	p@5	p@10	r@1	r@5	r@10	f1@10
SVDFeature	0.0085	0.0131	0.013	0.0023	0.0188	0.0371	0.0192
IRenMF	0.0209	0.0234	0.0243	0.006	0.0335	0.0698	0.0360
Rank-GeoFM	0.0209	0.0278	0.0273	0.0058	0.0387	0.0763	0.0403
Event2Vec-1	0.1778	0.1237	0.0922	0.0463	0.1581	0.2312	0.1318
Event2Vec-2	**0.2006**	**0.1350**	**0.1014**	**0.0514**	**0.1715**	**0.2522**	**0.1447**
Event2Vec-3	0.1561	0.1099	0.0829	0.0403	0.1401	0.2097	0.1188
Dataset	Douban-bej						
Metric	p@1	p@5	p@10	r@1	r@5	r@10	f1@10
SVDFeature	0.0382	0.0296	0.026	0.0073	0.0267	0.0468	0.0334
IRenMF	0.0323	0.0311	0.0297	0.0069	0.0287	0.0502	0.0373
Rank-GeoMF	0.0344	0.0353	0.0326	0.007	0.0318	0.0543	0.0407
Event2Vec-1	0.244	0.1658	0.1275	0.0409	0.1284	0.1866	0.1515
Event2Vec-2	**0.2572**	**0.1748**	**0.1312**	**0.0451**	**0.1431**	**0.2055**	**0.1602**
Event2Vec-3	0.1154	0.0772	0.0571	0.0226	0.0726	0.1031	0.0735
Dataset	Douban-sha						
Metric	p@1	p@5	p@10	r@1	r@5	r@10	f1@10
SVDFeature	0.0456	0.0328	0.0269	0.0183	0.0631	0.1009	0.0425
IRenMF	0.0656	0.0533	0.0436	0.0284	0.1031	0.1568	0.0683
Rank-GeoFM	0.0692	0.0567	0.0452	0.0297	0.1063	0.1596	0.0704
Event2Vec-1	0.1721	0.0988	0.0718	0.054	0.1342	0.1825	0.1031
Event2Vec-2	**0.2245**	**0.1215**	**0.0884**	**0.0763**	**0.1825**	**0.2516**	**0.1308**
Event2Vec-3	0.1124	0.0653	0.0459	0.0419	0.1108	0.1479	0.07

learning of the location and the time less distinguishable and results in a worse performance than other Event2Vec models.

4.3 Impact of Different Factors

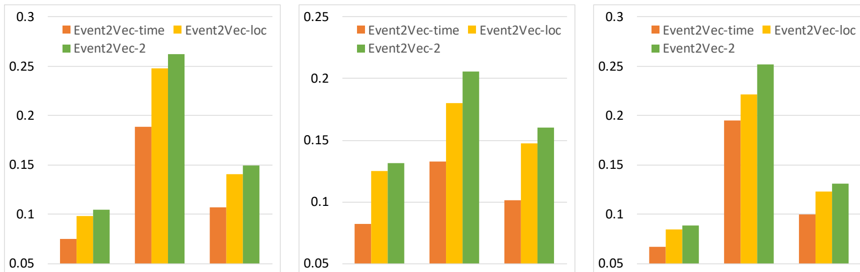
To explore the benefits of incorporating spatial and temporal influences into Event2Vec models respectively, we compare our Event2Vec model with two variants—Event2Vec-loc and Event2Vec-time. All three original Event2Vec models will reduce to the same architecture when only including one factor of the location or the time.

Event2Vec-time is the first simplified version where we ignore the spatial information in Event2Vec models.

Event2Vec-loc ignores the temporal information in Event2Vec models.

Event2Vec-2 is our best model by learning embeddings for spatial-temporal pairs.

We show the results on three datasets in Fig. 3. From Fig. 3, we first observe that Event2Vec-2 consistently outperforms the other two variants on all metrics, indicating



(a) Meetup (b) Douban-bej (c) Douban-sha

Fig. 3. The effect of different factors

Table 2. Comparison of Temporal Patterns

Dataset	Meetup			Douban-bej			Douban-sha		
Metric	p@10	r@10	f1@10	p@10	r@10	f1@10	p@10	r@10	f1@10
Weekday-Hour	0.1045	0.2627	0.1495	0.1312	0.2055	0.1602	0.0884	0.2516	0.1308
Day-Hour	0.0929	0.2355	0.1332	0.1037	0.1649	0.1274	0.0856	0.2575	0.1285
Month-Weekday-Hour	0.0893	0.2258	0.128	0.1208	0.19	0.1477	0.0901	0.2641	0.1343
Month-Day-Hour	0.0881	0.2239	0.1264	0.1129	0.1812	0.1391	0.086	0.2593	0.1292

that Event2Vec-2 takes advantage of both spatial and temporal influences simultaneously. Moreover, it's observed that the contributions of two factors to performance improvement are different. By comparing Event2Vec-time and Event2Vec-loc, we find that spatial influence is more significant than temporal influence for event recommendation.

4.4 Exploring Various Temporal Patterns

Our model recommends events to a user by taking advantage of the temporal influence. So far, we have evaluated its recommendation performance using a weekday-hour pattern, while its recommendation ability is not limited to one specific temporal pattern. By taking different definitions of temporal state, some other temporal patterns can be used for event recommendation with our model. For example, apart from the weekly pattern, we could also define the temporal state as daily pattern (day of the month); monthly pattern (month of the year); and their combinations. The only change made to our model is to divide time slots using different strategies. Table 2 shows the recommendation results of our model using different temporal patterns. The results show that the weekday-hour pattern achieves the best overall performance. By comparing the weekday-hour pattern and day-hour pattern, we observe that day of the week is more informative than day of the month, which indicates human behaviors exhibit stronger temporal cyclic patterns in a week than in a month (like working purpose on weekdays and entertainment purpose at weekends). However, the month-weekday-hour pattern and the month-day-hour pattern perform slightly worse than the weekday-hour pattern and the day-hour pattern on Meetup and Douban-bej dataset. Possible reasons could be that user's behaviors don't have strong patterns at month level and that adding monthly pattern additionally causes the sparsity issue to representation learning in time space.

5 Related Work

Event-based social networks (EBSN) have attracted much attention from research community. A great deal of research has been conducted on EBSNs. For example, Brown et al. [2] suggested that geographical closeness could influence the formation of online communities. Liu et al. [9] observed that 81.93% of event participations by a user are within 10 miles of his/her home location. Pham et al. [13] presented a graph-based model for event recommendation and Cheng et al. [4] developed a particular location recommendation method based on user preferences. Zhang et al. [20] used the location-based features for group recommendations in EBSN. Qiao et al. [15] proposes an approach to combine the heterogeneous social relationships, geographical features of events and implicit rating data from users to recommend events to users. However, most of these methods simply consider the spatial information as a bias factor and ignore the location-related semantic information.

From an algorithmic perspective, embedding techniques has been applied in a quantity of works such as network embedding [12], user profiling [16], social media prediction tasks [17], E-commerce product recommendation [14], and many other works. The embedding methods based on representing entries in low dimensional vector space, while preserving their properties, have been proved useful in multiple machine learning tasks such as classification, prediction and so on. However, no previous works have employed the representation learning methods in EBSN scenario where spatial and temporal factors have significant influences on user's behaviors.

6 Conclusion

In this paper, we study the recommendation problem in event-based social networks (EBSN). We proposed Event2Vec, a new embedding method that incorporates the spatial-temporal information jointly. We embed the event, location and time into low dimensional space based on event sequential data by taking advantages of the multitask learning and parameter sharing techniques. Different variants of Event2Vec are exploited to leverage the interactive influence between the spatial and temporal information.

We conducted extensive experiments to evaluate the performance of Event2Vec model on real-world datasets. The results showed superiority of our proposed model over other competitor methods. Moreover, we analyzed the effectiveness of spatial-temporal influences and compared different temporal patterns in user's behaviors in experiments.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI'16 (2016)

2. Brown, C., Nicosia, V., Scellato, S., Noulas, A., Mascolo, C.: Where online friends meet: Social communities in location-based networks. In: ICWSM. Citeseer (2012)
3. Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., Yu, Y.: Svdfeature: a toolkit for feature-based collaborative filtering. Journal of Machine Learning Research **13**(Dec), 3619–3622 (2012)
4. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: Successive point-of-interest recommendation. In: IJCAI'13. pp. 2605–2611 (2013)
5. Gao, H., Tang, J., Hu, X., Liu, H.: Exploring temporal effects for location recommendation on location-based social networks. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 93–100. ACM (2013)
6. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. CoRR **abs/1402.3722** (2014), http://arxiv.org/abs/1402.3722
7. Li, X., Cong, G., Li, X.L., Pham, T.A.N., Krishnaswamy, S.: Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp. 433–442. ACM (2015)
8. Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., Rui, Y.: Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 831–840. ACM (2014)
9. Liu, X., He, Q., Tian, Y., Lee, W.C., McPherson, J., Han, J.: Event-based social networks: Linking the online and offline social worlds. In: KDD'12. pp. 1032–1040 (2012)
10. Liu, Y., Wei, W., Sun, A., Miao, C.: Exploiting geographical neighborhood characteristics for location recommendation. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 739–748. ACM (2014)
11. Nielsen, L.: Personas-user focused design, vol. 15. Springer Science & Business Media (2012)
12. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
13. Pham, T.A.N., Li, X., Cong, G., Zhang, Z.: A general graph-based model for recommendation in event-based social networks. In: ICDE'15. pp. 567–578 (2015)
14. Phi, V.T., Chen, L., Hirate, Y.: Distributed representation based recommender systems in e-commerce. In: DEIM Forum (2016)
15. Qiao, Z., Zhang, P., Cao, Y., Zhou, C., Guo, L., Fang, B.: Combining heterogenous social and geographical information for event recommendation. In: AAAI'14. pp. 145–151 (2014)
16. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: ACL (1). pp. 1014–1023 (2015)
17. Wijeratne, S., Balasuriya, L., Doran, D., Sheth, A.: Word embeddings to enhance twitter gang member profile identification. arXiv preprint arXiv:1610.08597 (2016)
18. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 363–372. ACM (2013)
19. Zhang, W., Wang, J.: A collective bayesian poisson factorization model for cold-start local event recommendation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1455–1464. ACM (2015)
20. Zhang, W., Wang, J., Feng, W.: Combining latent factor model with location features for event-based group recommendation. In: KDD'13. pp. 910–918 (2013)