Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Who will Attend This Event Together? Event Attendance Prediction via Deep LSTM Networks

Xian Wu * Yuxiao Dong † Baoxu Shi * Ananthram Swami ‡ Nitesh V. Chawla *

Abstract

Event-based social network (EBSN) services have emerged as a new platform on which users can choose events of interest to attend in the physical world. Over years, there are growing research interests in predicting whether certain actors will participate in an event together. In this work, we refer to this task as the *event attendance prediction* problem and aim to address the predictability of individuals' event attendance. In real-world settings, the factors that influence an individual's attendance may change over time, leading to the dynamic nature of individuals' behavior. However, existing event attendance prediction methods cannot deal with such dynamic scenarios. To address this issue, we propose an end-to-end *Deep Event Attendance Prediction* (DEAP) framework—a three-level hierarchical LSTM architecture—to explicitly model users' multi-dimensional and evolving preferences. Extensive experiments on three real-world datasets demonstrate that DEAP significantly outperforms the state-of-the-art techniques across various settings.

1 Introduction

Event-based social networking platforms, such as Meetup and Plancast, have emerged as new paradigms, which enable users to link their online interactions with offline activities. In those EBSN services, groups host events and users can choose events (*e.g.*, parties and seminars) they are interested in to attend, facilitating face-to-face interactions in the physical world [22, 1]. Along this trend, one of the most important issues lies in the prediction of event attendees, which in turn benefits event recommendation for users [19].

Most of the existing event recommendation and attendance prediction techniques [19, 5, 7, 32] are proposed for static scenarios. They cannot handle the real-world scenarios where users' event preferences evolve over time. For example, users may prefer outdoor events during the summer while indoor ones during the winter. Predictive models trained in a static manner are not capable of capturing such dynamics of user preferences, motivating the necessity to keep the dynamic nature of users' event preferences in mind when developing actionable models and solutions.

In order to capture the users' evolving event-preferences, however, there exist several key challenges. i) Events in EBSNs are typically short-lived, making the inference of future attendees face the cold-start issue. That is, it is challenging to explore and leverage the trace of event's past participants for the future events. ii) The underlying factors that affect users' event participation may change over time. It is difficult to model this type of factor dependencies directly from users' sequential preferences. iii) Users' event attendance behaviors are closely related to the contextual information of events (*e.g.*, location and time of event). It is a non-trivial task to incorporate the context-aware constrains into predictive models. iv) The connections between event attendance can be arbitrary since any pair of events could potentially be related for various reasons. This introduces a significant challenge to handle all events with unknown dependencies.

To address the aforementioned challenges, this work develops an end-to-end framework—*Deep Event Attendance Prediction* (DEAP)—by explicitly modeling users' evolving preferences from the following three dimensions:

- *Sequential Preferences*: Users' historical attendance patterns could indicate their event preferences of different topics. In this work we use Sequential Preference to track users' interest-dependent attendance logs.

- *Contextual Preferences*: In addition to personal interests, users' participation behavior on EBSNs can also be influenced by their real-life events or experiences, which can be conceptually modeled by the spatial and temporal features. Herein we use such contextual signals to dynamically model the real-life influence through users' participation histories.

*University of Notre Dame. {xwu9,bshi,nchawla}@nd.edu
†University of Notre Dame & Microsoft. ydong1@nd.edu
‡Army Research Laboratory. ananthram.swami.civ@mail.mil

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

- *Exclusive Preferences*: In real life, both explicit and implicit influences between events are ubiquitous during users' decision-making processes. For example, users who participate in weekend events usually tend to group all events in a single day and skip others even if they are interested in them. In such cases, the attendance behaviors across events are no longer independent. Therefore, the future participants of an event can also be influenced by other events that share similar spatial or temporal characteristics.

To incorporate the above dimensions into event attendance prediction, our DEAP framework leverages a *three-level* hierarchical Long Short-Term Memory (LSTM) architecture. DEAP explores the rich contextual information of events to address the aforementioned event cold-start challenge. In specific, its first level transforms events' contextual information into latent embedding vectors in a non-linear way. In the second level, we aim to encode the evolving exclusive preferences of users by considering their attendance behavior across different groups. In DEAP's third level, it encodes users' sequential preferences to capture the time-evolving attendance patterns and interacts with the generated embedding vectors from the first two levels. With the three-level LSTM architecture, the generated semantic embedding vectors encode multi-dimensional preferences (*i.e.*, sequential, contextual, and exclusive preferences). Finally, the embeddings are fed into a Multilayer Perceptron (MLP) for predicting the event attendance of each user.

To demonstrate the effectiveness of the proposed DEAP method, we perform extensive experiments on three real-world datasets from Meetup, which represent a spectrum of geographical diversity. We compare our framework with state-of-the-art techniques and the evaluation results show that our scheme significantly outperform the baselines. In addition, we analyze the sensitivity of the proposed framework with respect to different model parameters.

In summary, our contributions are three-fold:

- To the best of our knowledge, this is the first work to study the event attendance prediction problem under dynamic scenarios, where users' event preferences evolve over time.
- We develop a Deep Event Attendance Prediction (DEAP) framework that leverages a three-level hierarchical LSTM architecture to capture users' multi-dimensional preferences (*i.e.*, sequential, contextual, and exclusive preferences).
- We perform various experiments on three real-world Meetup datasets to demonstrate the outperformance of the proposed DEAP framework by comparing with state-of-the-art techniques.

2 Problem Formulation

In this section, we begin with some necessary notations and then formally present the problem formulation. We consider a scenario with a set of users U (*i.e.*, U_1, U_2,...,U_M), a set of events E (*i.e.*, E_1, E_2,...,E_N) and a set of groups G (*i.e.*, G_1, G_2,...,G_K) which host events E. We refer to an individual user as $U_i \in U$, an individual event as $E_j \in E$ and an individual group as $G_k \in G$, where i, j and k are the user, event and group identifiers, respectively. In Meetup, a group can host different events at different times and each event can only be hosted by one group. Additionally, users can choose their interested events to attend and join the groups of like-minded people.

To deal with the cold-start problem of events as we introduced before, we associate each event with the notation relevant to its host group. In particular, each event $E_j \in E$ is associated with a group G_k and the sequence order information of all G_k's hosted events. Thus, we can denote E_j as E_k^t to represent the chronological t-th hosted event of group G_k. We use E_j and E_k^t interchangeably throughout this paper. We now formally define the user's evolving preferences from three dimensions which serve as inputs to our model:

DEFINITION 1. ***Sequential Preference Vector*** SP. *The Sequential Preference Vector SP is defined to represent the past attendance behavior of each user given a specific event. Specifically, given user U_i and the target event E_j (i.e., E_k^t) which is group G_k's t-th hosted event chronologically, $SP_{i,j}$ is a vector of length q (indexed by t') to indicate whether user U_i attended E_j's previous q events which are also hosted by group G_k. $SP_{i,j}^{t'} = 1, t' \in [1, ..., q]$ indicates that user U_i attended event $E_k^{t-t'}$ and $SP_{i,j}^{t'} = 0$ otherwise, where $E_k^{t-t'}$ represents group G_k's chronological (t-t')-th hosted event.*

DEFINITION 2. ***Contextual Preference Matrix*** CP. *The Contextual Preference Matrix CP is defined to capture the contextual signals (i.e., spatial-temporal constrains) in EBSNs data. Specifically, given user U_i and the target event E_j, $CP_{i,j}$ is a matrix of size $5 \times q$. Each column in $CP_{i,j}$ corresponds to each element in sequential preference vector $SP_{i,j}$ by including four key spatial-temporal features:*

- *spatial constrains: where event E_j will be held (i.e., the geographical coordinates of event E_j)*

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited
Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

- *temporal constrains: when event E_j will begin.*
  - **i)** *The day of a week (i.e., Mon-Sun).*
  - **ii)** *The hour of a day (i.e., 00-24).*
  - **iii)** *The time difference between two consecutive events ($E_k^{t-t'}$, $E_k^{t-t'+1}$) hosted by G_k.*

Definition 3. ***Exclusive Preference Vector*** EP. *The Exclusive Preference Vector EP is defined to indicate the influences among events. Given user U_i and target event E_j (E_k^t), each entry in $EP_{i,j}$ corresponds to each two consequent events of user U_i in $SP_{i,j}$ and $EP_{i,j}$ is a vector of size T. Specifically, we set the corresponding element in $EP_{i,j}$ as 1 if user U_i attended the event hosted by other group between event $E_k^{t-t'}$ and $E_k^{t-t'+1}$. Otherwise, we set the element as 0. Figure 1 shows an illustrative example for $EP_{i,j}$ vector given user U_i and event E_k^t.*

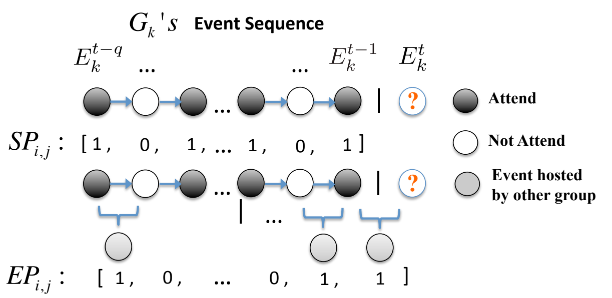


Figure 1: Illustrative example for exclusive preference vector $EP_{i,j}$ given user U_i and target event E_j (E_k^t).

Problem Statement. We define the user-event attendance matrix as $Y_{M\times N}$, in which $y_{i,j}$ represents the (i,j)-th entry of Y. We set the entry $y_{i,j} = 1$, if user U_i attended event E_j. Otherwise, $y_{i,j} = 0$. The objective of event attendance prediction is to learn a predictive model that explicitly exploits the above multi-dimensional factors (*i.e.*, SP, CP and EP) to infer future event attendance, i.e., unknown values in Y.

3 The Proposed DEAP Framework

In this section, we present the Deep Event Attendance Prediction (DEAP) framework to solve the event attendance prediction problem formulated in Section 2. We first give a quick review of the basic (Long Short-Term Memory) LSTM model and then present our solution in detail.

3.1 Basic LSTM. Recurrent Neural Network (RNN) has been widely used to address various challenges in time series analysis. LSTM is a special kind of RNN which captures long-term dependencies in recognizing time series patterns. We first describe how to apply the basic LSTM model to solve our problem and discuss its limitations.

LSTM proposes to derive the vector representation of hidden states h_t and c_t for each time step t as follows:

$$\begin{aligned}
i_t &= \sigma(W_i h_{t-1} + V_i x_t + b_i)\\
o_t &= \sigma(W_o h_{t-1} + V_o x_t + b_o)\\
f_t &= \sigma(W_f h_{t-1} + V_f x_t + b_f)\\
\widetilde{c_t} &= \phi(W_c h_{t-1} + V_c x_t + b_c)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}\\
h_t &= o_t \odot \phi(c_t) \qquad (3.1)
\end{aligned}$$

where $W_* \in \mathbb{R}^{d_s\times d_s}$ represents the transformation matrix from the previous state (*i.e.*, c_{t-1} and h_{t-1}) to LSTM cell and $V_* \in \mathbb{R}^{d_x\times d_s}$ are the transformation matrices from input to LSTM cell, where d_x and d_s denotes the dimension of input vectors and hidden states, respectively. Furthermore, $b_* \in \mathbb{R}^{d_s}$ is defined as a vector of bias term. $\sigma(\cdot)$ and $\phi(\cdot)$ represents the sigmoid and tanh function, respectively. The $\odot$ operator denotes the element-wise product. In Equation 3.1, i_t, o_t, and f_t represents input gate, output gate and forget gate, respectively. For simplicity, we denote Equation (3.1) as $[c_t, h_t] = \mathrm{LSTM}(x_t, c_{t-1}, h_{t-1})$ in the following sections.

Based on the above definitions, the event attendance prediction problem can be solved as follows: the model takes the past event attendance record of each user using one-hot encoding as input and outputs the hidden state in the corresponding time step. Then, the Multilayer Perceptron (MLP) component is utilized to derive the attendance probability by capturing the complex relationships between elements in hidden vectors. Formally, we represent MLP as follows:

$$\begin{aligned}
L_1 &= \phi_1(W_1 h_t + b_1)\\
&\ldots\ldots\\
L_n &= \phi_n(W_L L_{n-1} + b_L)\\
\hat{y}_{i,j} &= \sigma(W_y L_n + b_y) \qquad (3.2)
\end{aligned}$$

where n represents the number of hidden layers which is indexed by l. For the L_l layer, ϕ_l, W_l and b_l represent the activation function (i.e., *ReLU* function) of MLP layers, weight matrix and bias vector, respectively. We further specify the activation function as *sigmoid* (denoted as σ) to output the attendance probability. For simplicity, we denote Equation 3.2 as $\hat{y} = \mathrm{MLP}(h_t, n)$. In the experiments, we set the number of layers in MLP as 3.

However, two significant limitations exist in the basic LSTM model: (i) it ignores the contextual preferences, namely spatial-temporal signals in EBSNs, in modeling temporal evolution of user's preferences; (ii)

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

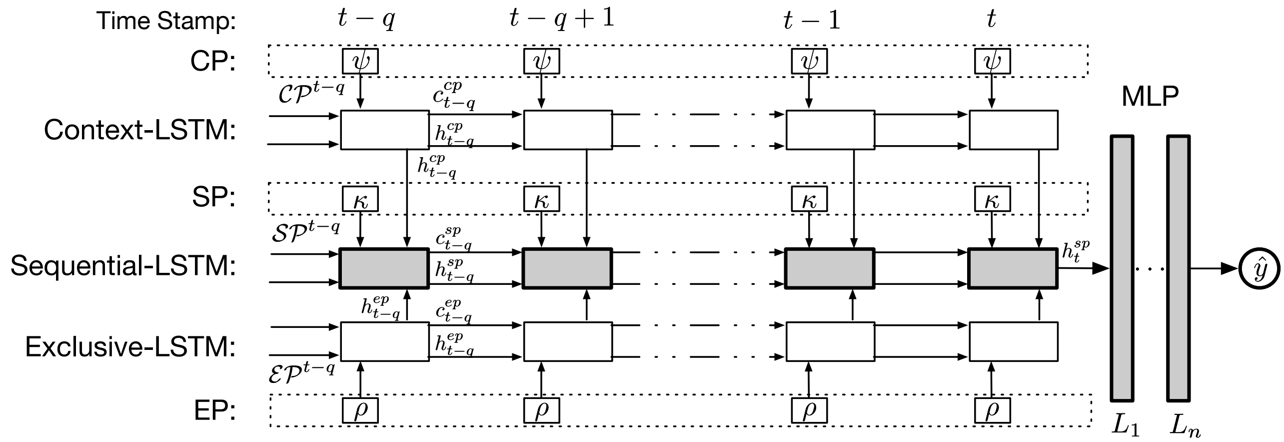


Figure 2: The Deep Event Attendance Prediction (DEAP) Framework. ψ and κ generate the corresponding contextual preference matrix and sequential preference vector for some given event E, ρ produces the exclusive preference vector given an event set within time span $\delta(t-1, t)$ and does not contain events from G_k.

it assumes that events are independent and ignores the fact that the user event attendance behaviors may be influenced by those of others. Hence, we need to develop a new model to explicitly consider these two important pieces of information in predicting future event attendances of users.

3.2 The DEAP Framework. To address the above limitations of the basic LSTM model, we present our DEAP framework to predict future event attendance of users by exploring users' evolving preferences from three dimensions (*i.e.*, Sequential Preference, Contextual Preference and Exclusive Preference). The proposed DEAP not only explores the long dependencies but also characterizes the semantic information in modeling the temporal dynamics within users and events. In particular, DEAP is a hierarchical LSTM framework consisting of four major components: *Sequential-LSTM Encoder*, *Context-LSTM Encoder*, *Exclusive-LSTM Encoder* and *MLP*, as shown in Figure 2.

3.2.1 Context-LSTM Encoder. To encode the evolving contextual preferences of users, Context-LSTM encoder generates contextual embedding without using hand-crafted features. Specifically, this encoder maps each contextual signal (*e.g.*, day of the week or time of the day) for user U_i and event E_j to an embedding. We formally define the corresponding hidden states c_t^{cp} and h_t^{cp} in encoding the contextual sequence as:

$$[c_t^{cp}, h_t^{cp}] = \mathrm{LSTM}(\mathcal{CP}^t, c_{t-1}^{cp}, h_{t-1}^{cp}) \tag{3.3}$$

where LSTM$(\cdot)$ represents the function in Equation 3.1 and $\mathcal{CP}^t$ denotes the vector of contextual preference matrix at time step t.

3.2.2 Exclusive-LSTM Encoder. To encode the evolving implicit influences among events, the Exclusive-LSTM encoder takes the element of the exclusive preference vector at each time step as input and updates its corresponding hidden states c_t^{ep} and h_t^{ep} as follows:

$$[c_t^{ep}, h_t^{ep}] = \mathrm{LSTM}(\mathcal{EP}^t, c_{t-1}^{ep}, h_{t-1}^{ep}) \tag{3.4}$$

where $\mathcal{EP}^t$ represents the element of the exclusive preference vector at t time step.

3.2.3 Sequential-LSTM Encoder. To encode the evolving sequential preferences of users, the Sequential-LSTM encoder integrates the past attendance behavior with the hidden states of other two encoders as follows:

$$[c_t^{sp}, h_t^{sp}] = \mathrm{LSTM}([\mathcal{SP}^t; h_t^{cp}; h_t^{ep}], c_{t-1}^{sp}, h_{t-1}^{sp}) \tag{3.5}$$

where c_t^{sp}, h_t^{sp} and $\mathcal{SP}^t$ denote the hidden states and the element of the sequential preference vector at t time step, respectively. $[\mathcal{SP}^t; h_t^{cp}; h_t^{ep}]$ represents the concatenation of the three vectors.

Output. Finally, with the above derived hidden states h_t^{sp} which capture the multi-dimensional evolving preferences of users, we can estimate $\hat{y}_{i,j}$ using MLP(h_t^{sp}, n) as we defined before.

3.3 Learning. In this subsection, we first describe the learning process of our DEAP framework. Then, we further utilize the advanced techniques (*i.e.*, *dropout* and *batch normalization*) to optimize the DEAP.

3.3.1 The Loss Function. As we introduced in the previous section, our objective is to derive the value of

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited
Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$y_{i,j}$ which denotes whether user U_i will attend event E_j or not. A commonly used metric in the loss function of binary classification task is *cross entropy* [26]. Thus, we define our loss function as follows:

$$\mathscr{L} = -\sum_U \sum_E y_{i,j} \ln \hat{y}_{i,j} + (1 - y_{i,j}) \ln(1 - \hat{y}_{i,j}) \tag{3.6}$$

where $\hat{y}_{i,j}$ denotes the estimated probability of user U_i attending event E_j. The weights can be achieved by minimizing the loss function.

3.3.2 Dropout. Overfitting has become one of the major issues for predictive analytics and especially for neural network models [27]. To address the overfitting issue in learning the weights, we utilize the dropout technique in the training process of our DEAP model. The basic idea of dropout is to randomly deactivate neurons in a layer with a certain probability drawn from the Bernoulli distribution [20] during training. In DEAP framework, we apply dropout by probabilistically excluding input and recurrent connections to LSTM units when training the nerual network. We formally apply the dropout in cell vector updating as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot D(\widetilde{c}_t, r) \tag{3.7}$$

where $D(\cdot)$ represents the dropout operation and r is the dropout ratio. We also apply dropout to Multilayer Perceptron component.

3.3.3 Batch Normalization. To overcome the observed covariance shift [25], in DEAP we use *Batch Normalization (BN)* to normalize the output of previous layers before sending it to the next one [12]. Such normalization has shown the ability of accelerating network training with a faster convergence rate and achieving higher accuracy on various tasks [9, 24, 30]. In DEAP framework, we apply *BN* to reduce the internal covariance shift by transforming the input to zero mean/unit variance distributions in each mini-batch training. It can be formally presented as:

$$\widetilde{c}_t = \phi(BN(W_c h_{t-1} + V_c x_t + b_c)) \tag{3.8}$$

where $BN(\cdot)$ stands for the batch normalization operation. *BN* is also applied to Multilayer Perceptron part.

4 Evaluation

In this section, we report on extensive experiments to evaluate the performance of the DEAP on real-world datasets collected from Meetup–an Event-based Social Network service. In particular, we answer the following research questions:

- **Q1:** How does DEAP perform as compared to the state-of-the-art solutions in predicting future event attendance of users?
- **Q2:** Does DEAP consistently outperform other algorithms in prediction accuracy with respect to geographical diversity (*i.e.*, datasets from different cities).
- **Q3:** How do different choices of parameters (*i.e.*, state size, embedding size, sequence length and dropout ratio) affect the performance of DEAP?

Table 1: The Statistics of Datasets

Datasets	Portland	Phoenix	Chicago
# of Users	14,372	18,068	30,608
# of Events	11,151	14,719	18,630
# of Groups	421	544	867
# of Event Attendance	67,245	102,812	143,040
Attendance Density	0.042%	0.039%	0.025%

In the following subsections, we present the data statistics, and experiment settings and followed answers to above research questions.

4.1 Experimental Setup.

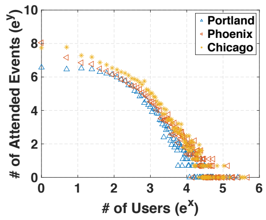
4.1.1 Datasets. In our evaluation, we perform experiments on Meetup datasets. Meetup enables a new type of social platform by integrating conventional online social networks with offline social interactions. In Meetup, users can join different groups and groups can create the events in which other users can participate. Each user's event attendance is formatted as: (user ID, event ID, group ID, timestamp). We targeted three cities (*i.e.* Portland, Phoenix and Chicago) in the U.S for evaluations. These cities are selected for two main reasons: (i) Different urban scales, reflective of different event attendance activities. (ii) Those cities are located in different states, which represents diversified attendance preferences. Table 1 summarizes the statistics of these data traces. Additionally, we plot the statistics on event attendance over time in Figure 3(c) to show that participation behaviors change from month to month.

4.1.2 Parameter Settings. We summarize the parameter settings of DEAP in Table 2. In addition, we vary each key model parameter in DEAP and fix others to examine parameter sensitivity. We choose Adam [13] as our optimizer to train the model.

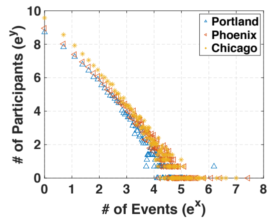
4.1.3 Evaluation Protocols. In our evaluation, we split the datasets chronologically into training (10 months), validation (half month) and test (1.5 months) sets. We use the validation datasets to tune hyper-parameters and test datasets to evaluate the final performance of all compared algorithms. Furthermore, our event attendance prediction problem suffers from

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

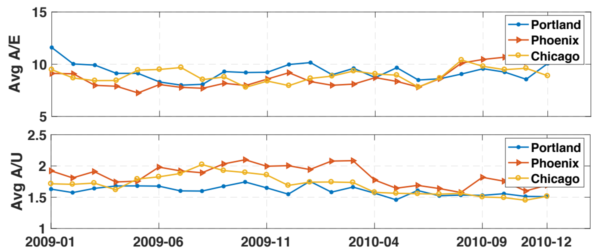
Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Distribution of Users' Attended Events



(b) Distribution of Events' Participants



(c) Event Attendance Distributions over Time (Avg A/E: Average Number of Participants for each event; Avg A/U: Average Number of Attended Events for each user.)

Figure 3: Event Attendance Distributions on Meetup

Table 2: Parameter Settings

	Parameter	Value
Model Parameter	Size of hidden state vector	32
	Embedding size of feature representation	8
	Sequence size of evolving preferences	7
	Dropout Ratio	0.5
Experimental Parameter	Scale parameter of BN component	0.001
	Shift parameter of BN component	0.99
	Batch size	50
	Learning rate	0.001

highly unbalanced data (e.g., with only 0.025% positive instances and 99.975% negative instances in Chicago datasets), as shown in Table 1. To address this challenge, we leverage the undersampling technique which has been widely applied to deal with imbalanced data [2]. In particular, given an event E_j in the training process, we generate training instances by sampling all its participants U_{E_j} and randomly sampling its non-participants with size of $2 \cdot |U_{E_j}|$.

To evaluate the performance, we adopt *Macro-F1* and *Micro-F1* metrics, which have been widely used for evaluating classification algorithms [8]. The reason we choose those metrics in this work is: there exists significant differences among the number of participants of different events, as we shown in Figure 3. Table 3 summarizes the mathematical definitions of the above metrics. In the evaluation, the event attendances, which are classified as "attend" or not by a particular scheme correctly, are regarded as TP or TN, respectively. The behaviors of attend and not attend, which are

Table 3: Evaluation Metrics

Metric	Definition
Macro-F1	$\frac{2\cdot\sum_{j=1}^{N} TP_j}{2\cdot\sum_{j=1}^{N} TP_j+\sum_{j=1}^{N} FN_j+\sum_{j=1}^{N} FP_j}$
Micro-F1	$\frac{1}{N}\cdot\sum_{j=1}^{N}\frac{2\cdot TP_j}{2\cdot TP_j+FN_j+FP_j}$

misclassified as not attend and attend, are regarded as FN and FP, respectively. Note that a higher Macro-F1 and Micro-F1 score indicate better performance.

4.1.4 Baselines.

We implemented DEAP using TensorFlow. In our evaluation, we consider two types of baselines: (i) feature-based schemes for recommendations or predictive analytics, and (ii) variations of Recurrent Neural Network models for time series prediction which consider sequential patterns.

- *Tensor Factorization (TF)* [29]: it models user's event attendance with a three-dimensional tensor, where the three dimensions stand for users, groups and time slots.
- *Neural Network Matrix Factorization (NMF)* [6]: it aims to predict event attendances of users by integrating event features with the generated embedding vector of its corresponding group.
- *Deep Neural Network (DNN)* [3]: it incorporates the three-dimensional preferences, which are studied in this work, into a neural network architecture.
- *Long Short-Term Memory (LSTM)* [10]: it utilizes the basic LSTM model to predict users' event attendances by addressing the long-term dependency problem in attendance sequence.
- *Multi-Contextual Learning to Rank (MCLRE)* [19]: it proposes a hybrid scheme to estimate the likelihood of users to attend events by exploiting multiple contextual signals in EBSNs data.
- *Spatial-Temporal Recurrent Neural Networks (ST-RNN)* [17]: it is applied in event attendance prediction problem by modeling the time interval and geographical distance information into a Recurrent Neural Network architecture.

For the parameter settings in baselines: (i) we set the embedding size of TF and NMF as 32; (ii) we set the state size of LSTM and ST-RNN as 32; (iii) DNN and MCLRE methods share the same inputs as our DEAP framework.

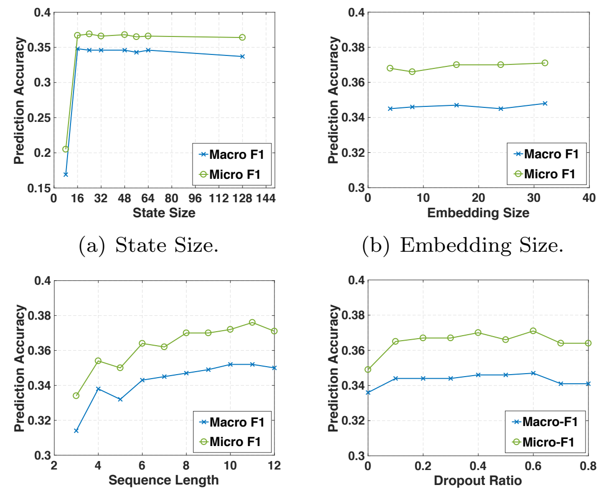
Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.2 Performance Comparison (Q1 and Q2): We now compare DEAP with state-of-the-art techniques as we introduced above. The evaluation results on Portland, Phoenix, and Chicago datasets are shown in Table 4, Table 5, and Table 6 respectively. To investigate the performance of all compared algorithms with different targeted time frames, we show the results from Jan-Feb to Sep-Oct in the table. From those evaluation results, we have the following two key observations.

First and foremost, we observe that DEAP consistently outperforms all compared baselines over different time frames. For example, on Portland, Phoenix and Chicago data traces, DEAP outperforms the best baseline (*i.e.*, ST-RNN) in terms of Macro-F1 and Micro-F1 by: 16.7% and 9% (Jan-Feb), 11.5% and 5.5% (May-Jun), 9.4% and 2.1% (May-Jun), respectively. The evaluation results across three cities with different time frames demonstrate the effectiveness of our DEAP framework in modeling evolving preferences of users. Second, we can observe that DEAP shows improvement over LSTM scheme on all datasets. This sheds light on the limitation of LSTM scheme which only model the sequential attendance behavior of users over time, and thus the usefulness of modelling contextual and exclusive preferences. Additionally, the large performance gap between DEAP and NMF indicates the limitation of NMF which ignores the evolving preferences (*e.g.* contextual constrains in events) of users.

4.3 Parameter Sensitivity (Q3). The DEAP algorithm involves several parameters (*i.e.*, state size d_s, embedding size, sequence length T and the dropout ratio r). To investigate the robustness of DEAP framework, we examine how the different choices of parameters affect the performance of DEAP in predicting event attendance of users. Except for the parameter being tested, we set other parameters at the default values (see Table 2). Figure 4 shows the prediction accuracy (measured by Macro-F1 and Micro-F1) as a function of each of the four parameters with the other three fixed. From Figure 4(a), we can observe that performance becomes stable as long as the state size is above 16. Similarly, we can observe that the parameter of embedding size has a relatively low impact on the performance of DEAP. Additionally, different from the weak effect of varying state size d_s and embedding size in prediction accuracy, step size q is positively correlated with the framework performance, as observed from Figure 4(c). In Figure 4(d), we can observe that the performance of our scheme shows an increasing trend with an increasing dropout ratio r for $r \leq 0.4$, while the performance is negatively correlated with r when $r > 0.6$. Because of space limitations, we only present the evaluation results



(a) State Size. (b) Embedding Size.

(c) Sequence Length. (d) Dropout Ratio.

Figure 4: Investigation of Parameter Sensitivity in Portland Data Trace

on the Portland data for Jan-Feb case. We also performed experiments to test the parameter sensitivity on Chicago and Phoenix, respectively. Similar results were observed.

5 Related Work

Event-based Social Networks (EBSNs) have received an increasing amount of attention in recent years and current research work has addressed various challenges in EBSN applications [18, 28, 23, 15]. For example, Li *et al.* defined a social event organization problem and assigned users to events so as to maximize the overall happiness of the users [15]. Liu *et al.* identified an event-based social network as co-existence between both online and offline social interactions [18]. A new arrangement scheme has been proposed to solve a more general event-participant arrangement problem by considering the conflicts of different events [23]. An emerging problem of event attendance prediction arises due to the rapid growth of event-based social network services (e.g., Meetup and Plancast). To address this emerging problem, this paper develops a novel framework to accurately predict the future event attendance of users by explicitly modeling their time-evolving preferences from three dimensions.

Our work is closely related to the works that address the problem of event recommendation and attendance prediction in EBSNs [19, 21, 7, 32, 4]. In particular, Gao *et al.* incorporated user preferences and social relationships into a event recommendation framework [7]. Macedo *et al.* proposed a multi-relational model to tackle the event recommendation problem [19].

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Table 4: Performance of all compared methods on Portland

Month	Jan-Feb		Mar-Apr		May-Jun		Jul-Aug		Sep-Oct	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
TF	0.222	0.255	0.211	0.249	0.212	0.256	0.194	0.227	0.206	0.250
NMF	0.178	0.236	0.177	0.192	0.204	0.259	0.193	0.171	0.115	0.184
DNN	0.207	0.269	0.163	0.242	0.093	0.169	0.162	0.273	0.106	0.200
LSTM	0.218	0.252	0.155	0.213	0.192	0.234	0.199	0.229	0.236	0.276
MCLRE	0.220	0.289	0.211	0.264	0.161	0.248	0.239	0.304	0.196	0.288
ST-RNN	0.313	0.337	0.323	0.356	0.306	0.346	0.290	0.320	0.284	0.327
DEAP	**0.346**	**0.367**	**0.339**	**0.375**	**0.357**	**0.377**	**0.332**	**0.336**	**0.338**	**0.367**

Table 5: Performance of all compared methods on Phoenix

Month	Jan-Feb		Mar-Apr		May-Jun		Jul-Aug		Sep-Oct	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
TF	0.218	0.387	0.210	0.329	0.226	0.373	0.187	0.272	0.187	0.274
NMF	0.176	0.218	0.151	0.159	0.190	0.236	0.165	0.199	0.116	0.192
DNN	0.088	0.190	0.163	0.273	0.212	0.342	0.114	0.227	0.127	0.226
LSTM	0.217	0.386	0.182	0.235	0.244	0.334	0.156	0.194	0.193	0.255
MCLRE	0.248	0.462	0.12	0.289	0.169	0.323	0.201	0.392	0.195	0.318
ST-RNN	0.295	0.484	0.300	0.425	0.329	0.457	0.299	0.448	0.252	0.354
DEAP	**0.342**	**0.508**	**0.322**	**0.439**	**0.360**	**0.467**	**0.323**	**0.460**	**0.281**	**0.374**

Table 6: Performance of all compared methods on Chicago

Month	Jan-Feb		Mar-Apr		May-Jun		Jul-Aug		Sep-Oct	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
TF	0.246	0.335	0.224	0.276	0.212	0.287	0.223	0.278	0.227	0.312
NMF	0.201	0.293	0.135	0.200	0.204	0.259	0.193	0.171	0.206	0.240
DNN	0.235	0.344	0.199	0.289	0.140	0.249	0.125	0.212	0.209	0.226
LSTM	0.246	0.340	0.226	0.263	0.208	0.254	0.224	0.305	0.182	0.296
MCLRE	0.283	0.398	0.186	0.271	0.148	0.235	0.227	0.319	0.250	0.368
ST-RNN	0.332	0.437	0.321	0.366	0.331	0.410	0.329	0.380	0.353	0.440
DEAP	**0.370**	**0.461**	**0.347**	**0.383**	**0.362**	**0.425**	**0.349**	**0.399**	**0.373**	**0.447**

Furthermore, Zhang *et al.* studied the problem of event attendance prediction based on spatial-temporal features [32]. Qiao *et al.* aimed at recommending events to users by modeling different sources (*e.g.*, geographical features and social relationships between users) into a Bayesian approach [7]. However, none of these techniques are proposed to work on dynamic scenarios. In contrast, this paper study the event attendance prediction problem by focusing on the challenge of evolving user preferences that have not been addressed by current solutions.

Our work is also related to the works on location recommendation and prediction in location-based social networks [31, 14, 16, 11]. For example, Zhang *et al.* developed a kernel density estimation method to recommend the Point-of-interest (POI) to user based on geographical proximity [31]. A geo topic model has been proposed for location prediction [14]. Furthermore, Lian *et al.* developed a regularized tensor factorization algorithm for temporal-aware location recommendation problem [16]. However, since location recommendation in location-based social network do not suffer from the cold-start problem of new item, these methods are not applicable.

6 Conclusion

We propose to predict event attendance of each user in a dynamic scenario, where user preferences evolve over time. To address this task, we develop a Deep Event Attendance Prediction (DEAP) framework which explicitly models evolving preferences of users from multi-dimensions. We evaluate our new solution on three real-world Meetup datasets. The experimental results demonstrate the effectiveness of our model and show that DEAP outperforms other state-of-the-art baselines in terms of prediction accuracy.

Acknowledgments

This work is supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, and NSF Grants IIS-1447795 and CNS-1622914.

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited

References

[1] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *KDD*, pages 114–122. ACM, 2011.

[2] N. V. Chawla, N. Japkowicz, and A. Kotcz. Special issue on learning from imbalanced data sets. *SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[3] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Recsys*, pages 191–198. ACM, 2016.

[4] Y. Dong, F. Pinelli, Y. Gkoufas, Z. Nabi, F. Calabrese, and N. V. Chawla. Inferring unusual crowd events from mobile phone call detail records. In *ECML/PKDD*, pages 474–492. Springer, 2015.

[5] R. Du, Z. Yu, T. Mei, Z. Wang, Z. Wang, and B. Guo. Predicting activity attendance in event-based social networks: Content, context and social influence. In *Ubicomp*, pages 425–434. ACM, 2014.

[6] G. K. Dziugaite and D. M. Roy. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*, 2015.

[7] L. Gao, J. Wu, Z. Qiao, C. Zhou, H. Yang, and Y. Hu. Collaborative social group influence for event recommendation. In *CIKM*, pages 1941–1944. ACM, 2016.

[8] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864. ACM, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] C. Huang, D. Wang, and S. Zhu. Where are you from: Home location profiling of crowd sensors from noisy and sparse crowdsourcing data. In *Infocom*, pages 1–9. IEEE, 2017.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *WSDM*, pages 375–384. ACM, 2013.

[15] K. Li, W. Lu, S. Bhagat, L. V. Lakshmanan, and C. Yu. On social event organization. In *KDD*, pages 1206–1215. ACM, 2014.

[16] D. Lian, Z. Zhang, Y. Ge, F. Zhang, N. J. Yuan, and X. Xie. Regularized content-aware tensor factorization meets temporal-aware location recommendation. In *ICDM*, pages 1029–1034. IEEE, 2016.

[17] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[18] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han. Event-based social networks: linking the online and offline social worlds. In *KDD*, pages 1032–1040. ACM, 2012.

[19] A. Q. Macedo, L. B. Marinho, and R. L. Santos. Context-aware event recommendation in event-based social networks. In *Recsys*, pages 123–130. ACM, 2015.

[20] D. Metzler, V. Lavrenko, and W. B. Croft. Formal multiple-bernoulli models for language modeling. In *SIGIR*, pages 540–541. ACM, 2004.

[21] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, L. Guo, and B. Fang. Combining heterogeneous social and geographical information for event recommendation. In *AAAI*, pages 145–151, 2014.

[22] J. She, Y. Tong, and L. Chen. Utility-aware social event-participant planning. In *SIGMOD*, pages 1629–1643. ACM, 2015.

[23] J. She, Y. Tong, L. Chen, and C. C. Cao. Conflict-aware event-participant arrangement and its variant for online setting. *TKDE*, 28(9):2281–2295, 2016.

[24] B. Shi and T. Weninger. Open-world knowledge graph completion. *arXiv preprint arXiv:1711.03438*, 2017.

[25] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[26] J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *TOIT*, 26(1):26–37, 1980.

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[28] Y. Tong, J. She, and R. Meng. Bottleneck-aware arrangement over event-based social networks: the max-min approach. *WWW*, 19(6):1151–1177, 2016.

[29] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, pages 211–222. SIAM, 2010.

[30] H. Zhang, M. Wang, R. Hong, and T.-S. Chua. Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing. In *MM*, pages 781–790. ACM, 2016.

[31] J.-D. Zhang and C.-Y. Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *SIGIR*, pages 443–452. ACM, 2015.

[32] X. Zhang, J. Zhao, and G. Cao. Who will attend?– predicting event attendance in event-based social network. In *MDM*, pages 74–83. IEEE, 2015.

Downloaded 05/16/18 to 131.107.147.65. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © 2018 by SIAM
Unauthorized reproduction of this article is prohibited