Spatiotemporal Activity Modeling via Hierarchical Cross-Modal Embedding

Yang Liu, Xiang Ao, *Member, IEEE*, Linfeng Dong, Chao Zhang, Jin Wang, and Qing He, *Member, IEEE*

Abstract—With the ever-increasing urbanization process, modeling people's spatiotemporal activities from their online traces has become a crucial task. State-of-the-art methods for this task rely on cross-modal embedding, which maps items from different modalities (e.g., location, time, text) into the same latent space. Despite their inspiring results, existing cross-modal embedding methods merely capture co-occurrences between items without modeling their high-order interactions. In this paper, we first construct two graphs from raw data records to represent the user interaction graph layer and activity graph layer and propose a hierarchical cross-modal embedding method that takes the high-order relationships into consideration. The key notion behind our method is a novel hierarchical embedding framework with meta-graphs connecting different layers. We introduce both *inter-record* and *intra-record* meta-graph structures, which enable learning distributed representations that preserve high-order proximities across graphs from different layers. Our empirical experiments on three real-world datasets demonstrate that our method not only outperforms state-of-the-art methods for spatiotemporal activity prediction, but also captures cross-modal proximity at a finer granularity.

Index Terms—Spatiotemporal activity, mobile data, cross-modal, hierarchical embedding

1 Introduction

With the rapid progress of urbanization [1], [2] worldwide, urban centres with large numbers of inhabitants are incessant to gather. According to the World Urbanization Prospects[1] published by the United Nations in 2018, the urban population of the world has increased to 4.2 billion, 55 percent of the world's population, in 2018 and by 2050, 68 percent of the world's population is projected to be urban. With such rapid urbanization process around the world, modeling people's activities has been recognized as an essential task [3] to handle with urban challenges like traffic congestion and resource allocation. Besides, choosing when and where to visit, eat or relax has become a fundamental demand for almost everyone, no matter local residents or ecdemic tourists. Answering questions like "Where should a shopping mania who cares about accessible transportation go?", "What are the popular activities around the beach at dusk?" and "When is the fit time for visiting the changing of the guard at the palace?" has become challenging not only for tourists, but even for local residents in the city because of their complex spatiotemporal dynamics.

Spatiotemporal activity modeling, which aims at modeling people's activities in different locations and time periods, plays an important role in solving these problems [3], [4]. The recent outgrowth of mobile data (e.g., geo-tagged social media, cellular data) sheds new light on automating this task. The number of worldwide mobile users has grown to 6.8 billion[2] and people can post their activities almost anytime and anywhere through their in-hand GPS-enabled mobile devices. Therefore, the mobile data provide an extensive and detailed coverage of urban activities, serving as a natural proxy for modeling human activities in urban spaces [5], [6], [7], [8].

The key to modeling spatiotemporal activities from mobile data is to define a cross-modal similarity that can capture the proximities between different modalities, e.g., location, time, and text. Most previous approaches exploit latent variable models for this problem [9], [10], [11], [12], [13], but such approaches are unscalable and rely on many prior distribution assumptions which may deviate from real data. Recently, cross-modal embedding methods [7], [14] have demonstrated inspiring results in this problem. Based on their co-occurrences within the same record, cross-modal embedding methods map items from different modalities into the same latent space to preserve their proximities.

Despite the remarkable success of existing cross-modal embedding techniques, they suffer from two major drawbacks in capturing item similarities. First, the interactions

1. https://population.un.org/wup/Publications/Files/WUP2018-KeyFacts.pdf

2. https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/

- Y. Liu, X. Ao, L. Dong, and Q. He are with the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: {liuyang17z, aoxiang, donglinfeng19s, heqing}@ict.ac.cn.
- C. Zhang is with the College of Computing, Georgia Tech, Atlanta, GA 30332. E-mail: chao.uiuc@gmail.com.
- J. Wang is with the Department of Computer Science, University of California, Los Angeles, Los Angeles, CA 90095. E-mail: jinwang@cs.ucla.edu.

Manuscript received 16 June 2019; revised 14 Jan. 2020; accepted 22 Mar. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Xiang Ao.)
Recommended for acceptance by L. Xiong.
Digital Object Identifier no. 10.1109/TKDE.2020.2983892

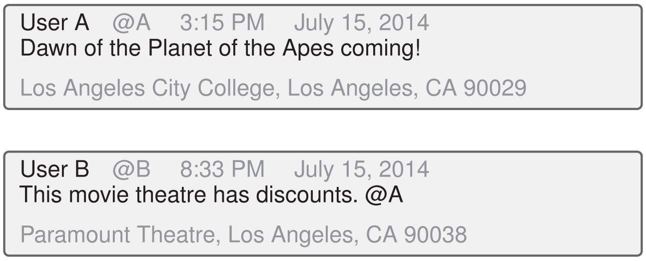


Fig. 1. Interactions between records of text-rich mobile data.

among items across different records are not adequately explored. For example, Fig. 1 demonstrates a pair of tweets, which is correlated through an "@" between two users. We can see User A is talking about a newly released movie, but the keywords are actually more related to the location and time specified by User B. Such proximities can be captured by high-order analysis of the information flow "text → user → user ⟨location, time⟩" across records, but would be missed if we only consider co-occurrences within single record. If we model user interactions at one layer and the activity at another layer, this kind of inter-record relationship will exist hierarchically across the two layers. Statistical data exhibit these inter-record interactions are prevalent in real-world corpus, e.g., 16.8 percent records have mentioned other users in UTGEO2011 dataset.[3] Taking such inter-record relationships into consideration may be useful for exploiting high-order information in the results of cross-modal predictions and facilitate the imperfects of previous alternatives.

Second, the semantics of the text intra the same record are not fully exploited. Existing methods usually regard each word as a basic textual unit and learn its embedding individually. However, it is known that the semantic meaning of a keyword depends on its context. As a result, conventional methods may suffer from the word sense disambiguation (WSD) problem since they fail to recognize context of keywords and capture the disambiguated meaning of them. For example, the keyword "ape" may indicate "imitate uncritically". But when surrounded by "drawn" and "planet", it should be recognized as "gorilla", and the phrase refers to a movie name. Therefore, the whole text message needs to be considered together when embedding the textual units of text-rich mobile data, which may enhance the performance of cross-modal embedding.

In this paper, we propose spatiotemporal activity modeling via hierarchical cross-modal embedding (ACTOR for short) from mobile data. Our method embeds items from different modalities (location, time, text) into a latent vector space, but differs from existing cross-modal embedding techniques in that it adopts a hierarchical embedding framework to preserve kinds of *high-order* item proximities. The hierarchy lies between the different constructed graphs from the raw mobile data. To fully encode the cross-modal co-occurrence relationship and user interactions, we first construct an activity graph and a user interaction graph, respectively. Then two kinds of meta-graphs, namely inter-record and intra-record meta-graphs are devised based on these two graphs to encode high-order relationships. Each graph acts as an embedding layer while nodes from different layers are embedded with the aid of meta-graphs. High-order proximity of vertices are preserved by the proposed meta-graphs because they include more than two pass-through hops in the graph. A hierarchical embedding framework is proposed based on meta-graphs which can preserve high-order proximities. Previous models could be considered as a single-layer special case of our framework.

We have performed experiments on three real-world datasets. The results demonstrate that the embeddings learned by ACTOR not only achieve the best quantitative performance in the cross-modal prediction tasks compared with the state-of-the-arts, but also preserve cross-modal proximities at a finer granularity. To the best of our knowledge, we are the first attempt to adopt hierarchical cross-modal embedding to model high-order information when modeling spatiotemporal activities.

The main contributions of this paper are highlighted as follows:

1) We propose a novel hierarchical cross-modal representation learning method for spatiotemporal activity modeling, which can preserve high-order proximities in mobile data. Different from previous studies, high-order information plays an important role in our embedding algorithm.
2) We propose a flexible meta-graph based embedding framework named ACTOR, which can perform hierarchical embedding on graphs of different layers. Specifically, we investigate several kinds of high-order meta-graphs in the proposed embedding algorithm.
3) We evaluate the effectiveness and efficiency of ACTOR on three real-world datasets. Experimental results demonstrate that ACTOR is a scalable framework and significantly outperforms the state-of-the-art methods in the tasks of cross-modal prediction and neighbor search.

The remainder of the paper is organized as follows. We summarize the related work in Section 2 and give the problem definition and overview in Section 3. Subsequently, graph construction and proximity are presented in Section 4. We introduce the framework of our method in Section 5, and the experimental results are shown in Section 6. We conclude this paper in Section 7.

2 Related Work

In this section, we briefly review the existing work related to our problem from the following three aspects: spatiotemporal activity modeling, graph representation learning and hierarchical graph embedding.

2.1 Spatiotemporal Activity Modeling

Spatiotemporal activity modeling has been receiving increasing research interest in the past few years. Existing methods can be categorized into two categories: topic model based and embedding based methods. Generally, the former extends classic topic models to bridge different data modalities, by assuming each latent topic can generate observations over not only textual keywords but also locations. [15] extends LDA by assuming multinomial distribution on text and Gaussian distribution over regions and [16] extends the model to more complex distributions. Kling *et al.* [17] extend

3. A large-scale worldwide tweet dataset created by mobile users.

PLSA with similar assumptions. One common limitation of the above methods is that they have to impose distribution assumptions on different modalities, which may not fit the true distribution in the real data well. Recently, embedding-based methods [7], [8], [14], [18], [19] have been proposed for spatiotemporal activity modeling. Zheng *et al.* [18] build a user-location-activity tensor and use factorization to learn latent representations for users and locations for personalized recommendation. Zhang *et al.* [7] propose a cross-modal embedding which maps different spatial units, temporal units and textual units into the same latent space to obtain their vector representations. Later on, they also develop a method [8] that processes continuous data streams and reveals recency-aware spatiotemporal activities. To address data scarcity problem, Zhang *et al.* design approaches [14] to transfer knowledge from external sources. Recently, some other researches focus on modeling sequential spatiotemporal activities, e.g., human flow prediction, etc. For example, Wang *et al.* [19] learn the representations from a flow graph and a spatial graph. Feng *et al.* [20] propose an attentional model named DeepMove to predict human mobility from the sparse and lengthy trajectories. Lin *et al.* [21] propose a deep learning-based convolutional model DeepSTN+ to predict crowd flows in the metropolis. Our work is related to [7] as we both use graph embedding for cross-modal representation learning. However, they do not consider high-order information like social relationship or semantic meaning.

2.2 Graph Representation Learning

Graph representation learning (also known as graph embedding) aims to learn low-dimensional representations for nodes or sub-graphs whose topological correlativeness in original graphs are preserved. Current methods can be categorized into random walk based and neural network based methods.

DeepWalk [22] is a representative homogeneous graph embedding method, which generalizes the skip-gram model in language modeling to graphs and exploits random walks to learn the features of vertices. Node2vec [23] investigates biased random walk to capture the diversity of connectivity patterns in networks. Tang *et al.* [24] introduce LINE, which defines loss functions to preserve the first-order and second-order proximity. Our work is different from DeepWalk, node2vec and LINE because they all belong to homogeneous graph embedding but the activity graph in this paper is a heterogeneous graph. Metapath2vec [25] is a recent representative heterogeneous graph embedding algorithm. It formalizes meta-path based random walks on the heterogeneous graph, which is not directly applicable for meta-graph based embedding in this paper.

Graph neural network [26], [27] is a series of neural network based graph representation learning methods. Graph convolutional neural network generalizes convolution operation to the graph domain, which can further be categorized as spectral approaches and spatial approaches. Spectral approaches [28], [29], [30], [31] work with a spectral representation of the graphs and the learned filters depend on the Laplacian eigenbasis. Spatial approaches [32], [33], [34] define convolutions directly on the graph. Our work is different from these graph neural network approaches since the main technical part of this paper belongs to random walk based methods. Therefore, we do not adopt neural network based methods as our baselines.

2.3 Hierarchical Graph Embedding

Recently, several attempts have been made to explore the hierarchical representations of nodes and graphs. For instance, Kriegel *et al.* [35] extend reference node embedding for approximating shortest path distance on graphs and propose hierarchical embedding to solve the problem of high storage cost. Mousavi *et al.* [36] propose a hierarchical framework which extracts local and global features from different scales of given graph at the same time. NetHiex [37] incorporates the hierarchical taxonomy into network embedding and HARP [38] decomposes a graph in a series of levels, and then embeds the hierarchy of graphs from the coarsest one to the original graph. DIFFPOOL [39] is a differentiable graph pooling module to generate hierarchical representations of graphs for the task of graph classification. Different from the above algorithms, the hierarchical learning process in this paper lies in modeling high-order relationships across or inside the records of text-rich mobile data, which are encoded by the proposed two kinds of meta-graphs.

3 Problem Definition and Overview

In this section, we give the description of mobile data and the problem definition of spatiotemporal activity modeling.

Let $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ be a corpus of mobile data records. Each record $r_i \in \mathcal{R}$ is defined by a tuple $\langle t_i, l_i, W_i \rangle$, $i = 1, 2, \ldots, N$, where

1) t_i is the creating timestamp of r_i;
2) l_i is a two-dimensional vector that represents the user's location when r_i is created;
3) $W_i = \{w_{i_1}, \ldots, w_{i_n}\}$ is a bag of keywords denoting the text message of r_i;

The problem of spatiotemporal activity modeling in this paper is to mine $\mathcal{R}$ and find some regularities of people's daily life. As there are three factors that are intertwined, an effective spatiotemporal activity model should accurately capture their cross-modal correlations. In another word, given any two of the three factors, the model is expected to predict the remaining one. Formally:

1) *Activity prediction.* Given t^*, l^* and a text candidate set $\mathcal{C}_w = \{w_1, \ldots, w_m\}$, find the most possible activity keyword w^* from $\mathcal{C}_w$;
2) *Location prediction.* Given t^*, W^* and a location candidate set $\mathcal{C}_l = \{l_1, \ldots, l_m\}$, find the most possible location l^* from $\mathcal{C}_l$;
3) *Time prediction.* Given l^*, W^* and a time candidate set $\mathcal{C}_t = \{t_1, \ldots, t_m\}$, find the most possible time t^* from $\mathcal{C}_t$.

An overview of the ACTOR framework could be found in Fig. 2. Hotspot detection is first conducted on the raw mobile data records and then we design two kinds of graphs to describe the data. After that, the hierarchical embedding algorithm could be applied on those graphs for downstream tasks like cross-modal prediction.

4 Graph Construction and Proximity

In this section, we first construct the activity graph and user interaction graph. Then we define proximity of different orders. Last, the algorithm for detecting spatial and temporal hotspots is introduced.

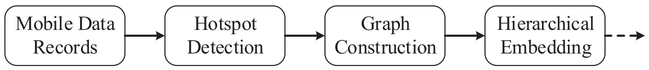


Fig. 2. The overview framework of ACTOR.

4.1 Activity Graph and User Interaction Graph

Definition 1 (Activity Graph). *An activity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a heterogeneous graph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of vertices including spatial, temporal and textual units, and $\mathcal{E}$ is a set of edges, $e_{ij} \in \mathcal{E}$ if and only if v_i and v_j appear in the same record, $i \neq j$, $i, j \in \{1, \ldots, n\}$. Moreover, $\mathcal{G}$ is associated with an vertex type mapping function $f_v : \mathcal{V} \rightarrow \mathcal{O}_v$ and an edge type mapping function $f_e : \mathcal{E} \rightarrow \mathcal{O}_e$, where $\mathcal{O}_v = \{T, L, W\}$ represents the vertex type set and $\mathcal{O}_e = \{TL, LW, WT, WW\}$ represents the edge type set. Within each edge type, the edge weight is set to be the co-occurrence count.*

Besides the co-occurrence of these units, mobile users often mention others in their own posts. Consequently, we can construct a user interaction graph to model this kind of behavior. Formally, we have the following definition.

Definition 2 (User Interaction Graph). *A user interaction graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a homogeneous graph, where $v_i \in \mathcal{V}'$ represents a mobile user and $e_{ij} \in \mathcal{E}'$ indicates that user i mentioned another user j, $i, j \in \{1, \ldots, |\mathcal{V}'|\}$. The edge weight is set to be the mentioned counts.*

As the example shown in Fig. 1, we can construct the corresponding activity graph and user interaction graph demonstrated in Fig. 3a. User B has mentioned user A in the textual records so there is an edge between user A and user B. The activity graph contains three modalities. The spatial unit comes from the location of the activity and the temporal unit derives from the created timestamp. These units are called spatial and temporal hotspots and the detection algorithm would be detailed in Section 4.3. The textual unit refers to the bag of words model in each record, where some frequent and meaningless words are removed. Since each co-occurrence appears only once, the weights of all edge are set to be 1 and we omit its weights for brevity.

4.2 Definition of Proximity

Based on a graph, we could define first-order proximity and second-order proximity. Furthermore, high-order proximity could also be introduced.

Definition 3 (First-order Proximity). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, \ldots, v_n\}$, the first-order proximity between a pair of vertices (v_i, v_j) is the edge weight if v_i and v_j are linked by an edge. If no edge is observed between v_i and v_j, their first-order proximity is 0.*

The neighborhood relationship in [7] stems from spatial and temporal continuities. Different from that, in this paper, we define the neighborhood relationship as a second-order proximity, which is widely used in network analysis [24], [40]. In another word, for any two vertices in our activity graph, the more neighbors they have in common, the more related they are.

Definition 4 (Second-order Proximity). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, \ldots, v_n\}$, the second-order proximity between a pair of vertices (v_i, v_j) is the similarity between their adjacency distribution, $i \neq j$, $i, j \in \{1, \ldots, n\}$. Mathematically, let $p_{v_i} = (a_{i1}, \ldots, a_{in})$ denote the first-order proximity of v_i, then the second-order proximity between v_i and v_j is determined by the similarity between p_{v_i} and p_{v_j}.*

In the activity graph, given a pair of vertices, the first-order proximity is defined to be the edge weight and the second-order proximity is the similarity between their adjacency distribution. High-order proximity is defined to be the connection with more than two hops in the graph. Taking Fig. 3a as an example, the temporal unit T_1 has high-order proximity with the textual unit W_2 via the connections in user interaction graph. We aim to design a hierarchical embedding framework with proximities of different orders preserved simultaneously.

4.3 Hotspot Detector

Due to the accuracy of the GPS-enabled devices and people's different customs and schedules, the raw mobile data displays obvious spatio-temporal variations and data sparsity. As addressed in [7], the spatial and temporal units in the activity graph of this paper comes from hotspot detection, since people's activities in urban areas often burst in geographical regions and time periods. Kernel density estimation is used to define the spatial and temporal hotspots since it has no assumption about the underlying data distribution. Suppose $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ are n data points in the d-dimensional space $\mathbb{R}^d$, the kernel density at any point $\mathbf{x}$ is given by

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right),$$

where $K(\cdot)$ is the Epanechnikov [41] kernel function and h is the kernel bandwidth.

Definition 5 (Spatial and Temporal Hotspots). *$\mathcal{R}$ is a mobile data corpus, $\mathcal{L}$ and $\mathcal{T}$ are the collections of locations and timestamps in $\mathcal{R}$, respectively. A spatial hotspot is defined as a local maximum of the kernel function estimated from $\mathcal{L}$. Similarly, a temporal hotspot is defined to be a local maximum of the kernel function estimated from $\mathcal{T}$.*

The mean shift [41] algorithm is employed to detect the spatial and temporal hotspots. For a given data point $\mathbf{x}$, which can be either location or timestamp, let $\mathbf{y}^{(k)}$ be the center of current window in iteration k, and $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be the m data points inside the window. The mean shift vector for $\mathbf{y}^{(k)}$ is $\mathbf{m}(\mathbf{y}^{(k)}) = \frac{\sum_{i=1}^{m}(\mathbf{x}_i - \mathbf{y}^{(k)})}{m}$, then $\mathbf{y}^{(k)}$ is shifted by $\mathbf{m}(\mathbf{y}^{(k)})$ as shown in Equation (1). The sequence $\{\mathbf{y}^{(k)}\}$ will converge to the hotspot that $\mathbf{x}$ belongs to. All the hotspots can be detected after performing this algorithm for every data point.

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{m}(\mathbf{y}^{(k)}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i. \quad (1)$$

After hotspot detection, for a new data point, we can find the hotspot that it belongs to by calculating the distances with all the detected hotspots and choosing the closest one.

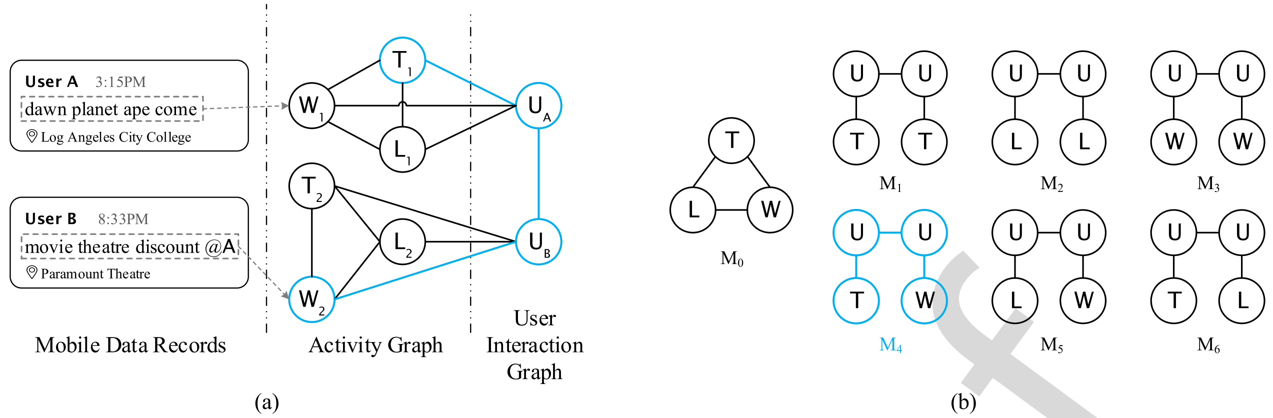


Fig. 3. (a) An illustrative example of the hierarchical embedding framework. T_i and L_i ($i = 1, 2$) are the spatial and temporal units derived from the timestamps and locations of the tweets. The textual units W_i ($i = 1, 2$) correspond to the words in the dashed box. Two units are connected if they appear in the same record. User B mentioned user A in the text thus the two users are linked. (b) The intra-record meta-graph M_0 are constructed according to the co-occurrence relationships of the spatial, temporal and textual units, which models high-order relationships inside records. The inter-record meta-graphs are built between the records who mentioned each other via the user interaction graph, which model high-order relationships between records. M_1 to M_6 are categorized according to different combinations of units connected to the users. The nodes and edges marked in blue color denote an instance of M_4.

5 The ACTOR Approach

In this section, we detail the proposed ACTOR approach. First, we introduce the definitions of the meta-graphs encoding inter-record and intra-record relationships of mobile data. Then we propose the hierarchical embedding framework based on the constructed graphs and proposed meta-graphs. Finally, we give the complete algorithm of ACTOR and some discussions about it.

5.1 Meta-Graph

Definition 6 (Meta-Graph). *A meta-graph $\mathcal{S} = (\mathcal{X}, \mathcal{A})$ is a sub-graphical scheme of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{X} \subseteq \mathcal{V}$ is a set of vertices along with its vertex type, and $\mathcal{A}$ is the adjacent relationship defined on $\mathcal{X}$.*

The intra-record meta-graph encodes the co-occurrence relationship inside the records shown as M_0 in Fig. 3b. The inter-record meta-graph aims to reflect the relationships among different records. According to different node types that it connects, it can be further categorized into M_1 – M_6 in Fig. 3b. It can be noticed that the inter-record meta-graphs depict high-order relationships between units in the activity graph since they contain more than two hops in the graph. For example, we can find an instance of M_4 in Fig. 3a.

5.2 Hierarchical Embedding Framework

Overall, as demonstrated in Fig. 3a, the framework contains three layers, the record layer, the activity graph layer and the user interaction graph layer. Correspondingly, the embedding framework can be decomposed into two steps. First, the user interaction graph is embedded to get the user embedding vectors from their interactive behaviors. Second, we devise a novel approach using meta-graph to model the high-order relationships between the user interaction graph and activity graph. The inter-record meta-graph connects two layers and guides the initialization of units in the activity graph from the embedding of user interaction graph. The intra-record meta-graph is employed to model the cross-modal co-occurrence relationship within the same record. The embedding objective is built based on both inter-record and intra-record meta-graphs.

5.2.1 Initialization

To begin with, the user interaction graph is embedded using LINE [24] and it is desired that those users who interact with each other frequently are close in the vector space. For those users who have never interacted with others, we use a random vector to represent them. The user embeddings are used to initialize the nodes in the activity graph. For a node in the activity graph, it may have connections with different users and we choose the user with the highest weight to get the initial embedding vector.

5.2.2 Embedding

Similar with the skip-gram [42] model, for each center vertex v_i and its known embedding vector $\boldsymbol{x}_i$, we want to predict the context embedding $\boldsymbol{x}'_j$ of its context vertex v_j. The context of v_i could be defined as all the v_j that $f_e(v_i, v_j)$ belongs to the same edge type, thus the context of a vertex may differ with different edge types. Given an edge type e and the center vertex v_i, the probability of context v_j generated by vertex v_i could be defined as Eq. (2).

$$p_e(v_j|v_i) = \frac{\exp({\boldsymbol{x}'_j}^{\mathrm{T}} \boldsymbol{x}_i)}{\sum_{f_e(v_i,v_k)=e} \exp({\boldsymbol{x}'_k}^{\mathrm{T}} \boldsymbol{x}_i)}, \tag{2}$$

$p_e(\cdot|v_i)$ defines a model distribution over the context of vertex v_i and the empirical distribution $\hat{p}_e(\cdot|v_i)$ could be defined by Eq. (3), where a_{ij} is the weight of the edge (v_i, v_j) and d_i^e is the degree of vertex v_i in the edge type e.

$$\hat{p}_e(v_j|v_i) = \frac{a_{ij}}{d_i^e}, \quad \text{where } d_i^e = \sum_{f_e(v_i,v_k)=e} a_{ik}. \tag{3}$$

To fully reconstruct the co-occurrence relationship, the conditional distribution of the contexts $p_e(\cdot|v_i)$ specified by the low-dimensional representation should be close to the empirical distribution $\hat{p}_e(\cdot|v_i)$. Therefore, we minimize the following objective function:

$$J_e = \sum_{v_i \in \mathcal{V}_e} \lambda_i D(\hat{p}_e(\cdot|v_i), p_e(\cdot|v_i)), \tag{4}$$

where $\mathcal{V}_e = \{v \in \mathcal{V} | \exists v' \in \mathcal{V}, \text{s.t. } f_e(v, v') = e\}$, λ_i is the importance weight of vertex v_i and $D(\cdot,\cdot)$ is the distance between two distributions. In this paper, we choose the KL-divergence as the measure between two distributions and evaluate the importance of vertex v_i by its degree d_i^e. In such settings, the objective function could be rewritten as

$$J_e = -\sum_{f_e(v_i,v_j)=e} a_{ij} \log p_e(v_j|v_i). \tag{5}$$

Since we have defined edge types and meta-graphs to preserve different orders of proximity, the overall objective function is

$$J = \sum_{e \in \mathcal{M}_{\text{intra}}} J_e + \sum_{e \in \mathcal{M}_{\text{inter}}} J_e, \tag{6}$$

where $\mathcal{M}_{\text{intra}} = \{\mathsf{TL}, \mathsf{LW}, \mathsf{WT}, \mathsf{WW}\}$ is the set of edge types in the intra-record meta-graph[4] and $\mathcal{M}_{\text{inter}} = \{\mathsf{UT}, \mathsf{UW}, \mathsf{UL}\}$ is part of the edge types in the inter-record meta-graph.

5.2.3 Optimization

When optimizing Eq. (5), the denominator in Eq. (2) requires the summation over all the edges of type e with center vertex v_i, which is highly computationally expensive. We adopt the approach of negative sampling proposed in [43]. Specifically, it specifies the following objective function for each edge (v_i, v_j):

$$J_{\text{NEG}} = -\log \sigma({\boldsymbol{x}'_j}^{\text{T}} \boldsymbol{x}_i) - \sum_{k=1}^{K} \mathbb{E}_{v_k \sim P(v)} \log \sigma(-{\boldsymbol{x}'_k}^{\text{T}} \boldsymbol{x}_i), \tag{7}$$

where σ is the sigmoid function. The first term models the observed edge (v_i, v_j) and the second term models the negative edges drawn from the noise distribution $P(v) \propto d_v^{\frac{3}{4}}$, where d_v is the out-degree of vertex v and K is the number of negative edges.

The updating rules for different variables can be derived by taking the derivatives of the above objective function and we list them as follows.

$$\frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}_i} = -[1 - \sigma({\boldsymbol{x}'_j}^{\text{T}} \boldsymbol{x}_i)] \boldsymbol{x}'_j + \sigma({\boldsymbol{x}'_k}^{\text{T}} \boldsymbol{x}_i) \boldsymbol{x}'_k \tag{8}$$

$$\frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}'_j} = -[1 - \sigma({\boldsymbol{x}'_j}^{\text{T}} \boldsymbol{x}_i)] \boldsymbol{x}_i \tag{9}$$

$$\frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}'_k} = \sigma({\boldsymbol{x}'_k}^{\text{T}} \boldsymbol{x}_i) \boldsymbol{x}_i. \tag{10}$$

The updating rule for edge type e can be written as Eq. (11).

$$\frac{\partial J_e}{\partial \boldsymbol{x}_i} = \sum_{f_e(v_i,v_j)=e} a_{ij} \frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}_i}. \tag{11}$$

Following [24], we sample from the original edges with sampling probabilities proportional to the original edge weights. Thus we could treat the weights of sampled edges as equal and choose a suitable learning rate η for the algorithm. The alias sampling [44] method is used for edge sampling, which takes $O(1)$ time when repeatedly drawing samples from the same distribution. We adopt the asynchronous stochastic gradient algorithm [45] for optimizing Equation (5). In each step, a mini-batch of edges from a certain kind of meta-graph are sampled, suppose the size of mini-batch is m, and the embedding vectors are updated by Equations (12), (13), and (14).

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \eta \sum_m \frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}_i} \tag{12}$$

$$\boldsymbol{x}'_j \leftarrow \boldsymbol{x}'_j - \eta \sum_m \frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}'_j} \tag{13}$$

$$\boldsymbol{x}'_k \leftarrow \boldsymbol{x}'_k - \eta \sum_m \frac{\partial J_{\text{NEG}}}{\partial \boldsymbol{x}'_k}. \tag{14}$$

5.3 ACTOR Algorithm

ACTOR is a hierarchical activity modeling framework based on mobile data generated in urban areas and the learning scheme of ACTOR is summarized in Algorithm 1.

Algorithm 1. ACTOR

Input: $\mathcal{R}$: A corpus of mobile data, $\mathcal{M}_{\text{inter}}$: inter-record meta-graphs, $\mathcal{M}_{\text{intra}}$: intra-record meta-graphs, d: The embedding dimension, K: Number of negative samples, $MaxEpoch$: Maximum iteration epochs, m: Number of sampling edges.

Output: The embedding vectors.

1: Apply the mean-shift algorithm to the timestamps and locations to detect spatial and temporal hotspots;
2: Construct an activity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, \ldots, v_n\}$ and a user interaction graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$;
3: Train the user interaction graph with LINE and get the user embeddings;
4: Initialize the center vectors $\{\boldsymbol{x}_i\}_{i=1}^n$ and context vectors $\{\boldsymbol{x}'_i\}_{i=1}^n$ of units in the activity graph with the corresponding pretrained user embedding vectors;
5: **for** $k = 0$ *to* $MaxEpoch - 1$ **do**
6:   **for** $e \in \mathcal{M}_{\text{inter}}$ **do**
7:     Sample m edges from $\mathcal{E}$ of type e;
8:     Updating $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{x}'_i\}$ with Equations (12), (13), and (14)
9:   **for** $e \in \mathcal{M}_{\text{intra}}$ **do**
10:     Sample m edges from $\mathcal{E}$ of type e;
11:     Updating $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{x}'_i\}$ with Equations (12), (13), and (14)
12: return $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{x}'_i\}_{i=1}^n$.

Given a corpus of mobile data $\mathcal{R}$, spatial and temporal hotspots are first detected (Line 1). After that, these hotspots, together with the textual units are constructed into an activity graph and a user interaction graph is built based on the mentioned records (Line 2). Then the user interaction graph is trained to get the user embedding vector (Line 3). For each vertex in the graph, we initialize its center vector and context vector with its pre-trained user embedding vector (Line 4). Then, we alternate the graph embedding

4. For the bag-of-words model in the intra-record meta-graph, we take the sum of all the textual unit embeddings in the same record.

TABLE 1
Statistics of Datasets

DATA	#Tweets	#Train	#Valid	#Test	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	#Spatial	#Temporal	#Word	#User
UTGEO2011	671,978	650,000	5,000	10,000	148,287	16,081,265	8,946	34	20,000	119,307
TWEET	1,188,405	1,000,000	20,000	50,000	174,578	28,521,412	10,420	27	20,000	144,131
4SQ	479,298	460,000	5,000	10,000	73,048	4,920,504	11,456	29	3,973	57,590

method to the instances of inter-record and intra-record meta-graphs and update the center and context vectors in the iteration (Line 5-11). Finally, the output is the center vectors and context vectors of all the vertices (line 12).

5.4 Discussions

We argue that ACTOR is a high-order proximity preserved cross-modal embedding algorithm. The inter-record meta-graph encodes the high-order proximity from activity graph to the user interaction graph since each instance of them contains more than two hops in the graph. The hierarchical embedding framework tends to preserve high-order proximity in the embedded space as encoded by the meta-graphs.

Besides, ACTOR is a general hierarchical cross-modal embedding framework, where meta-graphs can be flexibly assigned to probe connections between different graphs. Thus previous methods could be considered as special cases of ACTOR. For instance, CrossMap [7] could be obtained by embedding only the activity graph without hierarchical embeddding strategy.

Next we analyze the time complexity of the proposed ACTOR. Suppose d is the dimension of embedding vector and K is the number of negative samples, each step of optimization takes $O(d(K+1))$, under the condition that sampling an edge from the alias table takes constant time. And the iteration step is usually proportional to the number of edges $O(|\mathcal{E}|)$. Therefore, the overall time complexity of our proposed ACTOR is $O(dK|\mathcal{E}|)$.

6 Experiment

In this section, we report our experimental results on qualitative and quantitative evaluations of ACTOR on three real-world datasets.

6.1 Experimental Setup

6.1.1 Datasets

We conducted the experiments on three public benchmark datasets.

- UTGEO2011 [46] contains 38 million tweets collected across the entire globe between September 4th and November 29th, 2011. A subset is provided in [46] with around 10,000 users and we adopt it as benchmark dataset in our paper.
- TWEET [7] consists of 1.1 million geo-tagged tweets published in Los Angeles during August 1st to November 30th, 2014.
- 4SQ [7] includes around 0.6 million Foursquare checkins posted in New York from August 2010 to October 2011.

The train/valid/test split is done randomly from all the records and the detailed statistics of the datasets can be summarized in Table 1, including the scale of the corresponding constructed activity graphs.

6.1.2 Compared Methods

- *LGTA* [17] can discover and compare geographical topics from GPS-associated documents, combining both location and text information.
- *MGTM* [16] is a state-of-the-art geographical topic model which captures dependencies between geographical regions based on a multi-Dirichlet process.
- *Metapath2vec* [25] is a state-of-the-art heterogeneous embedding algorithm. It performs heterogeneous random walks on the graph according to the predefined meta-paths and then encodes each vertex into vector space.
- *LINE* [24] defines loss function to preserve the first-order or second-order proximity separately for graph embedding. We also adapt LINE to the activity graph with the auxiliary vertex type of U and derive LINE (U) as another baseline.
- *CrossMap* [7] is a state-of-the-art method for spatiotemporal activity modeling. It jointly maps different units into the latent space but only models the co-occurrence and neighborhood relationships. Similar as LINE(U), we also extend CrossMap on the activity graph with the auxiliary vertex type of U and derive CrossMap(U) for a comprehensive comparison.
- *ACTOR*: the model proposed in this paper.

6.1.3 Parameter Settings

The major parameters of ACTOR include the latent embedding dimension d, learning rate η, number of negative samples K, the batch size m, the maximum epoch $MaxEpoch$. For the three datasets above, we set $d = 300$, $\eta = 0.02$, $K = 1$, $m = 256$, $MaxEpoch = 100$. For the baselines, we finely tuned the corresponding parameters in order to perform a fair comparison. In our experiments the reported results are the average of five runs.

The ACTOR algorithm is implemented in C++ and experiments are conducted on a CentOS 6.9 server, with 32 cores Intel(R) Xeon(R) 2.10 GHz CPU and 64 GB memory.

6.2 Cross-Modal Prediction

6.2.1 Prediction Method

We quantitatively evaluate the performance of ACTOR by cross-modal prediction. It can be decomposed into three sub-tasks: activity prediction, location prediction and time prediction.

TABLE 2
Mean Reciprocal Rank for Cross-Modal Retrieval

Data	UTGEO2011			TWEET			4SQ		
Task	Text	Location	Time	Text	Location	Time	Text	Location	Time
LGTA	0.3571	0.3440	/	0.4615	0.4439	/	0.5739	0.5409	/
MGTM	0.2993	0.3022	/	0.3615	0.3619	/	0.4538	0.4191	/
metapath2vec	0.5062	0.5267	0.3169	0.5083	0.5369	0.2986	0.8475	0.8673	0.3262
LINE	0.5433	0.5442	0.3427	0.6246	0.5997	0.3235	0.9076	0.8954	0.3637
LINE(U)	0.5830	0.5798	0.3578	0.6315	0.6066	0.3297	0.9078	0.8972	0.3719
CrossMap	0.5778	0.6015	0.3852	0.6701	0.6561	0.3439	0.9393	0.9138	0.3690
CrossMap(U)	0.5808	0.6070	0.3712	0.6894	0.6632	0.3469	0.9441	0.9137	0.3735
ACTOR	**0.6207**	**0.6275**	**0.3885**	**0.6991**	**0.6805**	**0.3509**	**0.9519**	**0.9211**	**0.3758**

Take the location prediction as an example. Suppose we have obtained vector representations for all the units in the training corpus. For each query in the test set, with the time and text modalities known, the location candidate set is composed of the ground truth location and noisy locations that are randomly chosen from the spatial hotspots of the test set. Then we could compute the cosine similarity of each candidate location to the observed timestamp and keywords and rank them in the descending order in terms of similarity. The ranked list is regarded as the predicted result. In our experiments, besides the ground truth, 10 noisy candidates are randomly chosen from the test corpus and hence the size of candidate set is 11.

6.2.2 Evaluation Metric

The Mean Reciprocal Rank (MRR) is adopted to quantify the performance of this model. Formally, given a set Q of queries, the MRR is the average of the reciprocal ranks of each query in Q, as Eq. (15) shows.

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}, \tag{15}$$

where rank_i refers to the rank position of the ground truth for the ith query. Specifically, in this paper, each record in the test corpus is a query and rank_i refers to the rank position of the ith record.

6.2.3 Experimental Results and Discussions

The experimental results of various methods on the three datasets are presented in Table 2. For each dataset, we demonstrate the MRR metrics on three prediction tasks. From the table, we observe that ACTOR consistently outperform all the other methods on the two datasets, with at most 85.9 percent improvements compared with LGTA and 16.0 percent improvements with CrossMap.

LINE and metapath2vec are two representative graph embedding algorithms. LINE is designed mainly for homogeneous graph thus it performs much more poorly than ACTOR in embedding activity graph which contains vertices of different types. Metapath2vec is developed for heterogeneous graph but the embeddings rely on the generated random walks. We have tried to use the proposed meta-graphs $M_1 - M_6$ as meta-paths to generate random walks but the predict results are far from satisfactory. It is difficult to perform random walk on the user interaction graph since it is rarely sparse. Therefore, we explore other meta-paths and report the best scores on these three datasets in Table 2. The meta-path for UTGEO2011 and TWEET is $\mathsf{L}-\mathsf{W}-\mathsf{T}-\mathsf{W}$ but for 4SQ, both $\mathsf{L}-\mathsf{W}-\mathsf{T}-\mathsf{W}$ and $\mathsf{T}-\mathsf{L}-\mathsf{W}-\mathsf{W}$ are adopted. The window size and number of negative samples are set to be 3 and 5 respectively.

LINE and CrossMap are not originally designed for high-order embedding but they could be simply modified and applied on the activity graph with auxiliary vertex type U, which are the results of LINE(U) and CrossMap (U). Compared with LINE and CrossMap, the user vertices bring extra information and obtain performance improvements to some extent. However, through hierarchical cross-modal embedding, ACTOR could encode high-order proximities into the embedding procedures and consequently ACTOR performs better than LINE(U) and CrossMap(U).

6.2.4 Case Study

To figure out the reason why ACTOR outperforms other baselines, especially CrossMap, we perform cross-modal prediction on the same record and noise candidates using these two methods, then observe their ranking results.

For activity prediction task, the original tweet is shown as Fig. 4. The tweet was posted at a prop room while the attached text directly mentioned it. The aim is to tell the most possible text from the mix of 1 groundtruth and 10 randomly chosen noise text. The ranking results of ACTOR and CrossMap are presented in Fig. 5. As we can see, the groundtruth text ranked 1[st] in ACTOR but 7th in CrossMap. The hierarchical embeddings adopted by ACTOR could capture the cross-modal correlation precisely thus it can match the text with the location closely.

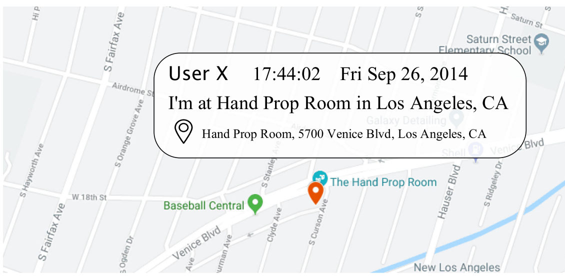


Fig. 4. The ground truth tweet for activity prediction.

Tweets	ACT	CM
I'm at Hand Prop Room in Los Angeles, CA	1	7
Break legs, @trippster88!!! I love yeww @ Rogue Machine Theatre	2	3
Follow @thebuzzeronfox. (@ FOX Sports Interactive Media in Los Angeles, CA)	3	5
Brunch'n in LA with Etuajie! No mimosas today, but still good. #labrunch #frenchtoast @ The Bossy…	4	1
dis new young toilet prod by polo club the most fire @johnassembly	5	9
#bts scene pick of a Lil old school #glam #Hollywood action for today's #musicvideo with the very…	6	2
Just watched a screening of The Judge for SAG voters and what a treat at the end	7	6
#Lakers #GoLakers LA Lakers Rumors: Michael Beasley, Jordan Crawford, Chris Singleton Free	8	8
#NOM @ Pickwick Gardens	9	4
METROPOLITAN FASHION WEEK #jasonryan #surlounge #metropolitanfashionweek	10	10
#Utilities #Job in #LongBeach, CA: Satellite TV Technician/Installer -- Long Beach, CA Area at DISH	11	11

Fig. 5. Ranking results of both methods for activity prediction. ACT is short for ACTOR and CM is short for CrossMap.

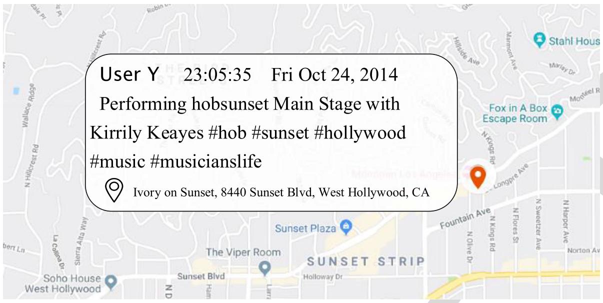


Fig. 6. The ground truth tweet for time prediction.

TABLE 3
Ranking Results of Both Methods for Time Prediction

Timestamps	ACTOR	CrossMap
Fri Oct 24 23:05:35 CDT 2014	1	7
Mon Oct 13 20:57:17 CDT 2014	2	3
Thu Aug 14 20:34:31 CDT 2014	3	5
Sat Aug 16 21:51:02 CDT 2014	4	1
Mon Aug 25 21:57:48 CDT 2014	5	9
Wed Aug 13 01:14:54 CDT 2014	6	2
Tue Oct 14 01:17:35 CDT 2014	7	6
Fri Oct 24 19:06:56 CDT 2014	8	8
Mon Aug 11 10:26:08 CDT 2014	9	4
Wed Nov 12 15:40:06 CST 2014	10	10
Wed Aug 20 11:47:08 CDT 2014	11	11

For time prediction task, the original tweet is shown as Fig. 6. The task is to predict the most possible time when the performance took place at this music bar. As Table 3 shows, the top 3 timestamps both methods returned are acceptable since most bars will arrange their performance at night, when the number of customers reaches the peak of a day.

For location prediction task, the original tweet was posted at a pavilion as Fig. 7 shows, which can be

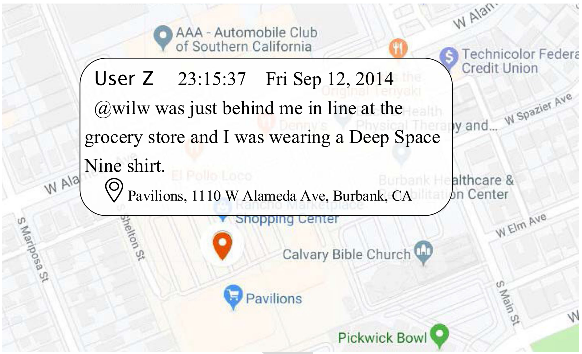
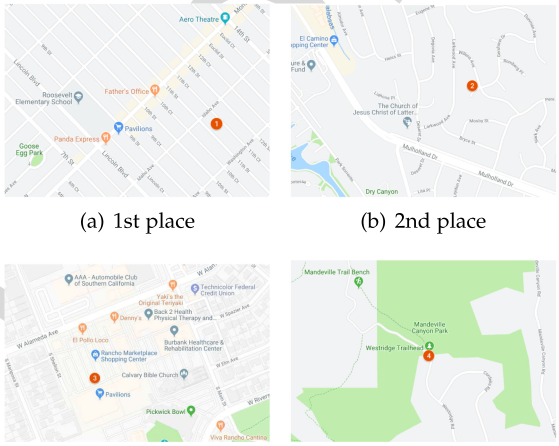


Fig. 7. The ground truth tweet for location prediction.



(a) 1st place (b) 2nd place

(c) 3rd place (d) 4th place

Fig. 8. Ranking results of CrossMap for location prediction.

inferred from the text as well. ACTOR ranked the groundtruth in the 1st place. The top 4 places that CrossMap returned are listed in Fig. 8, where the groundtruth was in the 3rd place. Although we can find another pavilion near the 1st place, there is no obvious connection between grocery store and the 2nd place, neither the 4th place. We infer that ACTOR could capture the function of the place due to multiple orders of proximities preserved in the activity graph while CrossMap may have some inaccurate correlations.

6.3 Ablation Test

We identify two key structures in our proposed ACTOR framework: inter-record structure and intra-record structure. Inter-record structure refers to the hierarchical embedding framework induced by the inter-record meta-graph, say the pre-training of user interaction graph and embedding with $\mathcal{M}_{\text{inter}} = \{\mathsf{UT}, \mathsf{UW}, \mathsf{UL}\}$ in the activity graph. Intra-record structure refers to the bag of words model in the intra-record meta-graph, that we consider words together rather than treat them as individual textual unit. We address the model without inter-record structure as ACTOR w/o inter, the model without intra-record structure as ACTOR w/o intra, and the complete model proposed in this paper as ACTOR-complete. The ablation test results can be found in Table 4.

As demonstrated in the table, both inter and intra structures of ACTOR contribute to the final performance. No

TABLE 4
Mean Reciprocal Rank for Ablation Test

Data	UTGEO2011			TWEET			4SQ		
Task	Text	Location	Time	Text	Location	Time	Text	Location	Time
ACTOR w/o inter	0.6040	0.6025	0.3723	0.6930	0.6742	0.3498	0.9492	0.9148	0.3754
ACTOR w/o intra	0.6072	0.6104	0.3628	0.6904	0.6635	0.3481	0.9443	0.9137	**0.3765**
ACTOR-complete	**0.6207**	**0.6275**	**0.3885**	**0.6991**	**0.6805**	**0.3509**	**0.9519**	**0.9211**	0.3758

matter which part of the model is removed, the MRR metric would decline a little. For UTGEO2011, hierarchical embedding strategy and inter-record meta-graph contribute more than intra-record meta-graph since the performance of ACTOR w/o inter is worse than ACTOR w/o intra. For TWEET and 4SQ dataset, we have no information about the user interactions but we can still link the units in the activity graph to the user and part of the inter-record meta-graph could also help with the cross-modal correlation as we can conclude from the results of ACTOR w/o inter for TWEET and 4SQ.

6.4 Neighbor Search

Next, we investigate the effectiveness of the obtained embeddings by qualitative comparisons. In particular, we evaluate the resultant cross-modal correlations through the results under different kinds of queries, namely spatial query, temporal query and textual query, on the TWEET dataset. From the previous comparison, CrossMap is shown as the strongest competitor, hence we focus on comparing our ACTOR and CrossMap in such evaluation.

6.4.1 *Spatial Query*

Fig. 9 shows the results when we query the location of the port of Los Angeles, whose latitude and longitude is (33.7395, -118.2599). The results of ACTOR are closely related to the port, like "dock", "departure" or the place "port of LA". However, CrossMap prefers some general words like "today", "time", etc. Clearly, ACTOR performs better in capturing the characteristic of the place than CrossMap.

6.4.2 *Temporal Query*

Fig. 10 shows the results of the temporal query of "10:00pm". From the figure, we observe both methods return temporal hotspots close to 10:00pm but the textual results differ a lot. Unlike CrossMap returns some general words like "tonight" or "like", ACTOR finds some specific activities in the evening, like listening to music,[5] watching TV series,[6] sports programs,[7] and some information about the occurring places, e.g., "dance hall" or "box seat". The results also demonstrate that ACTOR might correlate more specific activities.

ACTOR		CrossMap	
Text	Time	Text	Time
port of la	10:57:39	today	10:57:39
dock	14:34:54	day	17:42:27
groovecruise	17:42:27	time	14:34:54
departure	18:53:55	get	18:53:55
mex	10:13:51	camera	10:13:51
passport	10:38:16	work	13:33:17
berth	6:06:47	another	16:49:07
ship	16:49:07	segundo	15:51:17
segundo	14:59:13	got	10:38:16
evo	5:47:58	hit	14:59:13

Fig. 9. Spatial query of port of Los Angeles.

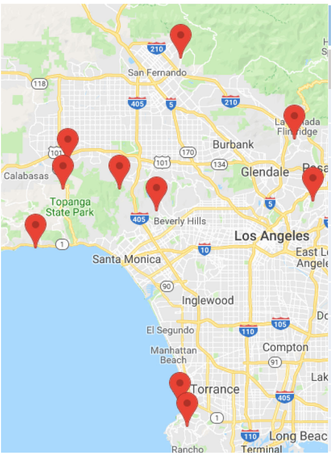
6.4.3 *Textual Query*

For the textual query, we search the popular sports bar "Patrick Molloy's Sports Pub" at Hermosa Beach, LA. The keyword for this bar is "patrick_molloy_sport_pub" in our vocabulary and the search results are shown in Fig. 11. Both methods return temporal hotspots around free time and spatial hotspots near the pub except one outlier in the result of CrossMap, but the textual results[8] differ. It is worth mentioning that ACTOR returns several specific words containing hermosa beach in which the pub is located at while CrossMap just returns similar pubs. Clearly, ACTOR embeds more information from the whole text than CrossMap.

6.5 Scalability

We finally evaluate the scalability of ACTOR on the TWEET dataset as we expand the sampling edges or increase the computing threads. The basic number of sampling edges is 4 million. First, we investigate the performance of ACTOR by multipling the sampling edges 1, 2, 3, 4 times and the total running time is shown in Fig. 12a, from which we argue that ACTOR is robust in dealing with increasing sampling edges as the running time scales linearly with the number of sampling edges. To study the strong scalability of ACTOR, we keep the basic number of sampling edges and vary the computing thread from 1 to 4. Fig. 12b exhibits the corresponding results. From the figure we argue that ACTOR is highly parallelizable using multi-thread stochastic gradient algorithm. To test the weak scalability, we keep the threads and edges growing in pace with each other and the performance is shown as Fig. 12c, the running time remains nearly constant as the simultaneous increase of both threads and edges. From the results we can conclude that ACTOR achieves a good scaleup. To sum up, the proposed ACTOR demonstrates a good scalability and is practical for large-scale datasets.

5. Ricky Martin is one of the iconic figures of the Latin American music scene.
6. *Masters of Sex* is an American period drama television series, the second season of which first aired on July 13, 2014 and last aired on September 28, 2014, receiving critical acclaim on Rotten Tomatoes and Metacritic.
7. Jim Fox is a Canadian retired former professional ice hockey player who played nine seasons in the NHL for the Los Angeles Kings. Now he is one of the analysts of FOX Sports West's Kings.
8. American Junkie, Baja Sharkeez, Abigaile are all sports bars.



ACTOR		CrossMap	
Text	Time	Text	Time
rickymartin	21:56:40	hiii	21:56:40
hiii	22:35:27	going	22:56:42
mastersofsex	22:56:42	tonight	20:52:57
jimfox19	0:19:12	like	22:35:27
california_losangeles_usa	1:07:47	get	1:07:47
dancehall	0:44:25	one	0:19:12
box_seat	17:42:27	time	18:53:55
straighten	18:53:55	dinner	19:47:05
mai	5:47:58	know	0:44:25
westridge	10:57:39	kinkyboots	17:42:27

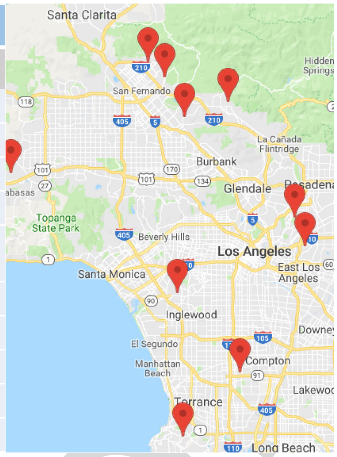
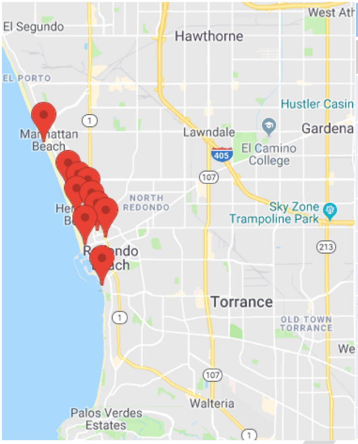


Fig. 10. Temporal query of 10:00pm.



ACTOR		CrossMap	
Text	Time	Text	Time
patrick_molloy_sport_pub	22:56:42	patrick_molloy_sport_pub	0:19:12
junkiehb	0:19:12	junkiehb	22:56:42
american_junkie	22:35:27	american_junkie	1:07:47
sharkeez	16:49:07	sharkeez	16:49:07
abigaile	17:42:27	abigaile	14:59:13
hermosabeach	19:47:05	campo	19:47:05
hermosa_beach_2nd_street	13:33:17	killershrimp	22:35:27
finfest	14:34:54	beachbums	13:33:17
saintrocke	11:18:04	hennessey	21:56:40
campo	14:59:13	nikkisvenice	17:42:27

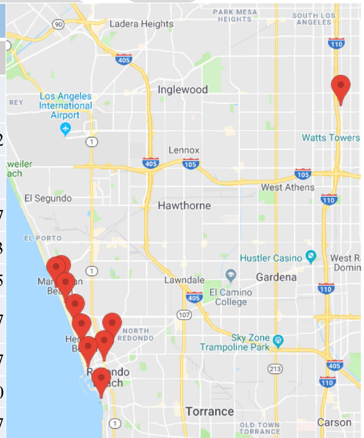
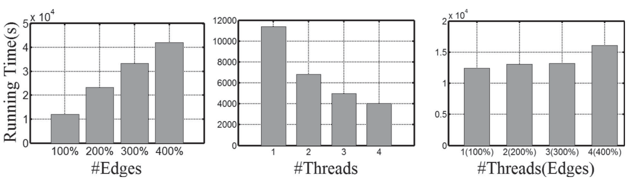


Fig. 11. Textual query of “patrick_molloy_sport_pub”.

7 Conclusion

In this paper, we study the problem of spatiotemporal activity modeling and propose ACTOR, a hierarchical cross-modal embedding framework. The key technical contribution lies in the design of meta-graphs for hierarchical embedding to capture high-order relationship of spatiotemporal activities. Combined with these meta-graphs, ACTOR jointly embeds all spatial, temporal and textual units into the same space where proximities of different orders are simultaneously probed. We conduct extensive experiments on three real-world datasets. The empirical results demonstrate that ACTOR significantly outperforms other baselines due to the preserved high-order proximities.



(a) Sizeup (b) Speedup (c) Scaleup

Fig. 12. Scalability of ACTOR.

Acknowledgments

The research work was supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002104, the National Natural Science Foundation of China under Grant No. 61976204, U1811461, the Project of Youth Innovation Promotion Association CAS. This work was also supported by the Natural Science Foundation of Chongqing under Grant No.cstc2019jcyj-msxmX0149. We thank the anonymous reviewers and the associated editor for their reviewing efforts.

References

[1] D. E. Bloom, D. Canning, and G. Fink, “Urbanization and the wealth of nations,” *Science*, vol. 319, pp. 772–775, 2008.

[2] H. Ritchie and M. Roser, “Urbanization,” *Our World Data*, 2019. [Online]. Available: https://ourworldindata.org/urbanization

[3] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, “Urban computing: Concepts, methodologies, and applications,” *ACM Trans. Intell. Syst. Technol.*. vol. 5, 2014, Art. no. 38.

[4] Y. Zheng, “Methodologies for cross-domain data fusion: An overview,” *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[5] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, “Who, where, when and what: Discover spatio-temporal topics for Twitter users,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 605–613.

[6] W. Kang *et al.*, “Trendspedia: An internet observatory for analyzing and visualizing the evolving Web,” in *Proc. IEEE 30th Int. Conf. Data Eng.*, 2014, pp. 1206–1209.

[7] C. Zhang *et al.*, “Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning,” in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 361–370.

[8] C. Zhang *et al.*, “ReAct: Online multimodal embedding for recency-aware spatiotemporal activity modeling,” in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 245–254.

[9] Q. Mei, C. Liu, H. Su, and C. Zhai, “A probabilistic approach to spatiotemporal theme pattern mining on weblogs,” in *Proc. 15th Int. Conf. World Wide Web*, 2006, pp. 533–542.

[10] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, “SeMiTri: A framework for semantic annotation of heterogeneous trajectories,” in *Proc. 14th Int. Conf. Extending Database Technol.*, 2011, pp. 259–270.

[11] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, “Semantic trajectories: Mobility data computation and annotation,” *ACM Trans. Intell. Syst. Technol.*, vol. 4, 2013, Art. no. 49.

[12] P. Wang, P. Zhang, C. Zhou, Z. Li, and G. Li, “Modeling infinite topics on social behavior data with spatio-temporal dependence,” in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1919–1922.

[13] F. Wu, Z. Li, W.-C. Lee, H. Wang, and Z. Huang, “Semantic annotation of mobility data using social media,” in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1253–1263.

[14] C. Zhang, M. Liu, Z. Liu, C. Yang, L. Zhang, and J. Han, “Spatiotemporal activity modeling under data scarcity: A graph-regularized cross-modal embedding approach,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 531–538.

[15] S. Sizov, “GeoFolk: Latent spatial semantics in web 2.0 social media,” in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 281–290.

[16] C. C. Kling, J. Kunegis, S. Sizov, and S. Staab, “Detecting non-gaussian geographical topics in tagged photo collections,” in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 603–612.

[17] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, “Geographical topic discovery and comparison,” in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 247–256.

[18] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, “Towards mobile intelligence: Learning from GPS history data for collaborative recommendation,” *Artif. Intell.*, vol. 184-185, pp. 17–37, 2012.

[19] H. Wang and Z. Li, “Region representation learning via mobility flow,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 237–246.

[20] J. Feng *et al.*, “DeepMove: Predicting human mobility with attentional recurrent networks,” in *Proc. World Wide Web Conf.*, 2018, pp. 1459–1468.

[21] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1020–1027.

[22] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[23] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: Large-scale information network embedding,” in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[25] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 135–144.

[26] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 729–734.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan, 2009.

[28] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in *Proc. Int. Conf. Learn. Representations*, 2014.

[29] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” 2015, *arXiv:1506.05163*. [Online]. Available: https://arxiv.org/pdf/1506.05163.pdf

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[31] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Representations*, 2017.

[32] D. K. Duvenaud *et al.*, “Convolutional networks on graphs for learning molecular fingerprints,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[33] J. Atwood and D. Towsley, “Diffusion-convolutional neural networks,” in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2001–2009.

[34] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[35] H.-P. Kriegel, P. Kröger, M. Renz, and T. Schmidt, “Hierarchical graph embedding for efficient query processing in very large traffic networks,” in *Proc. Int. Conf. Sci. Statist. Database Manage.*, 2008, pp. 150–167.

[36] S. F. Mousavi, M. Safayani, A. Mirzaei, and H. Bahonar, “Hierarchical graph embedding in vector space by graph pyramid,” *Pattern Recognit.*, vol. 61, pp. 245–254, 2017.

[37] J. Ma, P. Cui, X. Wang, and W. Zhu, “Hierarchical taxonomy aware network embedding,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1920–1929.

[38] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, “HARP: Hierarchical representation learning for networks,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2127–2134.

[39] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, “Hierarchical graph representation learning with differentiable pooling,” in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.

[40] D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1225–1234.

[41] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/pdf/1301.3781.pdf

[43] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

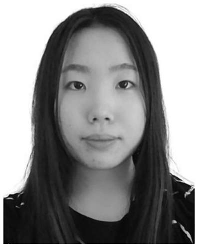
[44] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, “Reducing the sampling complexity of topic models,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 891–900.

[45] B. Recht, C. Re, S. Wright, and F. Niu, “HOGWILD: A lock-free approach to parallelizing stochastic gradient descent,” in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 693–701.

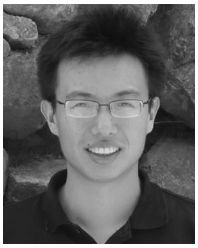
[46] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge, “Supervised text-based geolocation using language models on an adaptive grid,” in *Proc. Joint Conf. Empir. Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 1500–1510.

Yang Liu received the BS degree in mathematics from Nanjing University, Nanjing, China, in 2017. He is currently working toward the PhD degree in the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China. His research interests include graph representation learning, data mining for spatiotemporal activity modeling, and financial user modeling.

Xiang Ao (Member, IEEE) received the BS degree in computer science from Zhejiang University, Hangzhou, China, in 2010, and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2015. He is an associate professor of the Institute of Computing Technology, Chinese Academy of Sciences(ICT, CAS), Beijing, China. His research interests include user modeling and natural language processing for finance/business-related applications. He has authored more than 30 referred publications at prestigious conferences and journals like the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Intelligent Systems and Technology*, WWW, ICDE, SIGIR, IJCAI, EMNLP, etc.



Linfeng Dong received the bachelor's degree in computer science from the University of Chinese Academy of Sciences, Beijing, China, in 2019. She is currently working toward the master's degree of Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her research interests include graph representation learning and spatiotemporal data mining.



Chao Zhang received the PhD degree in computer science from the University of Illinois at Urbana-Champaign, Champaign, Illinois. He is an assistant professor with the College of Computing, Georgia Institute of Technology, Atlanta, Georgia. His research area is data mining and machine learning. He is particularly interested in developing label-efficient and robust learning techniques, with applications in text mining, and spatiotemporal data mining.

Jin Wang received the master's degree in computer science from Tsinghua University, Beijing, China, in 2015. He is currently working toward the PhD degree in Computer Science Department, University of California, Los Angeles, Los Angeles, California. His research interests include text analysis and processing, stream data management, and database system.

Qing He (Member, IEEE) received the BS degree from Hebei Normal University, Shijiazhuang, China, in 1985, the MS degree from Zhengzhou University, Zhengzhou, China, in 1987, both in mathematics, and the PhD degree in fuzzy mathematics and artificial intelligence from Beijing Normal University, Beijing, China, in 2000. He is a professor as well as a doctoral tutor with the Institute of Computing Technology, Chinese Academy of Science (CAS), Beijing, China, and he is a professor with the University of Chinese Academy of Sciences (UCAS), Beijing, China. His interests include data mining, machine learning, classification, and fuzzy clustering.