

Learning Urban Community Structures: A Collective Embedding Perspective with Periodic Spatial-temporal Mobility Graphs

PENGYANG WANG and YANJIE FU, Missouri University of Science and Technology

JIAWEI ZHANG, Florida State University

XIAOLIN LI, Nanjing University, Nanjing, China

DAN LIN, Missouri University of Science and Technology

Learning urban community structures refers to the efforts of quantifying, summarizing, and representing an urban community's (i) static structures, e.g., Point-Of-Interests (POIs) buildings and corresponding geographic allocations, and (ii) dynamic structures, e.g., human mobility patterns among POIs. By learning the community structures, we can better quantitatively represent urban communities and understand their evolutions in the development of cities. This can help us boost commercial activities, enhance public security, foster social interactions, and, ultimately, yield livable, sustainable, and viable environments. However, due to the complex nature of urban systems, it is traditionally challenging to learn the structures of urban communities. To address this problem, in this article, we propose a collective embedding framework to learn the community structure from multiple periodic spatial-temporal graphs of human mobility. Specifically, we first exploit a probabilistic propagation-based approach to create a set of mobility graphs from periodic human mobility records. In these mobility graphs, the static POIs are regarded as vertexes, the dynamic mobility connectivities between POI pairs are regarded as edges, and the edge weights periodically evolve over time. A collective deep auto-encoder method is then developed to collaboratively learn the embeddings of POIs from multiple spatial-temporal mobility graphs. In addition, we develop a Unsupervised Graph based Weighted Aggregation method to align and aggregate the POI embeddings into the representation of the community structures. We apply the proposed embedding framework to two applications (i.e., spotting vibrant communities and predicting housing price return rates) to evaluate the performance of our proposed method. Extensive experimental results on real-world urban communities and human mobility data demonstrate the effectiveness of the proposed collective embedding framework.

CCS Concepts: • **Information systems** → **Data mining; Location based services;**

Additional Key Words and Phrases: Urban communities, community structure, collective embedding, periodic mobility graphs

This research was partially supported by the National Science Foundation (NSF) via grant number 1755946, by the University of Missouri Research Board (UMRB) via proposal number 4991, and by the Natural Science Foundation of China (NSFC) via grant numbers 71701007 and 61773199.

Author's addresses: P. Wang and Y. Fu, Missouri University of Science and Technology, Rolla, MO, USA, 65409; emails: {pwqt3,fuyan}@mst.edu; J. Zhang, Florida State University, Tallahassee, FL, USA, 32306; email: jwzhang@cs.fsu.edu; X. Li, Nanjing University, Nanjing, China, 210023; email: lixl@nju.edu.cn; D. Lin, University of Missouri, MO, USA, 65211; email: lindan@missouri.edu. Yanjie Fu and Xiaolin Li are co-contact authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 ACM 2157-6904/2018/11-ART63 \$15.00

<https://doi.org/10.1145/3209686>

ACM Reference format:

Pengyang Wang, Yanjie Fu, Jiawei Zhang, Xiaolin Li, and Dan Lin. 2018. Learning Urban Community Structures: A Collective Embedding Perspective with Periodic Spatial-temporal Mobility Graphs. *ACM Trans. Intell. Syst. Technol.* 9, 6, Article 63 (November 2018), 28 pages.

<https://doi.org/10.1145/3209686>

1 INTRODUCTION

Learning urban community structures refers to the efforts of quantifying, summarizing, and representing a community's (i) static geographic structure, e.g., important Points-Of-Interests (POIs) and corresponding spatial allocations, and (ii) dynamic mobility structure, e.g., human mobility patterns among the important POIs. To be specific, the static geographic structure of a community refers to the spatial allocations and relative distances of the important POIs that provide a variety of urban functions [54] for the community. The dynamic mobility structure describes the strengths and dynamics of human mobility connectivity among these important POIs [59]. By learning the representation of community structures, we can better understand the evolution of urban communities over time. The knowledge and patterns obtained by analyzing urban community structures can be further used to help us find better solutions to boost commercial activities, enhance public security, foster social interactions, which will lead to livable, sustainable, and viable environments.

All the above pieces of evidence suggest that it is highly appealing to study how to quantify and discover urban community structures. Indeed, the emerging methodological studies on representation learning provide a great opportunity to address this problem. Inspired by the idea of representation learning, we propose to formulate the problem of learning urban community structures as a spatial representation learning task. Along this line, we develop a collective embedding analytic framework to learn urban community structures. The proposed collective embedding framework can unify both static POIs data and dynamic human mobility data as periodic spatiotemporal mobility graphs and collaboratively learn the embeddings of community structures from the spatial-temporal autocorrelations among multiple mobility graphs.

However, due to the complex nature of urban systems, urban community structure learning is not an easy task. Three unique challenges arise in achieving this goal:

- **Graph construction:** how to unify and represent the POIs and human periodic mobility records as a set of mobility graphs;
- **Collective embedding:** how to collectively learn the embeddings of POIs from multiple periodic mobility graphs;
- **Embedding aggregation:** how to align and aggregate POI embeddings for community structure representation learning.

In what follows, we outline how we tackle these challenges.

First, to carry out daily activities, people in urban areas often leave from one POI, visit another POI, and, thus, interact with communities. As a result, human movements create a dynamic association, which varies over time, between each POI pair in a community. Graphs are an effective tool to represent such a kind of structural information where we can regard POIs as vertexes, and treat mobility connectivities between POI pairs as edges. Then, given human periodic mobility records, we can construct a set of periodic spatial-temporal mobility graphs to capture the dynamics of a community. Unfortunately, since most of human mobility data are GPS data recorded by taxies, city bikes, subways, and buses, the pick-up and drop-off points are close to but not the exact origins and destinations of the travelers. For example, a person may take a short walk from his/her home to the bus stop while the dataset will only record the origin starting from the bus

stop instead of the home. To address this problem, we propose a probabilistic spatial propagation method to estimate mobility volumes between POI pairs.

Second, after representing human mobility records as graphs, an intuitive method is to exploit deep auto-encoder [6] to learn the embeddings of graph nodes (i.e., POIs). However, in this study, a community is represented by multiple periodic mobility graphs, which highly necessitates a new embedding method to simultaneously learn the embeddings of nodes (POIs) from multiple graphs in a collective fashion. Therefore, we develop a collective deep auto-encoder method that can take multiple graphs as inputs.

Third, after obtaining embeddings of each POI in the community, there is still a critical need to devise an effective fusion method to align and aggregate all the individual POI embeddings into the embedding of the community. We proceed with a two-step strategy. First, we use an unsupervised graph-based weighting method to compute the weight of each latent feature in the POI embeddings and then combine the weights to aggregate the embeddings of individual POIs to the embeddings of POI categories (e.g., education, shopping, restaurants, entertainment) that are semantically aligned across communities. Later, we further aggregate the embeddings of POI categories into the embeddings of the community.

To summarize, in this article, we propose a collective embedding framework to learn the community structure from the periodic spatiotemporal graphs of human mobility. Specifically, the followings are our four main contributions: (1) We start with a probabilistic propagation approach to construct a set of periodic mobility graphs to represent human periodic mobility records. (2) We propose a collective deep auto-encoder method to collaboratively learn the embeddings of POIs from multiple spatial-temporal mobility graphs. (3) Given the learned POI embeddings, we develop an unsupervised graph-based weighted aggregation approach to effectively align and aggregate the POI embeddings with the representation of the community structures. (4) We apply the proposed embedding framework to spot vibrant communities (i.e., urban vibrancy for short) and predict housing return rates (i.e., willingness to pay for short), and the extensive experimental results on real-world urban community and human mobility data demonstrate the effectiveness of our approach.

2 PROBLEM STATEMENT

In this section, we first introduce some important definitions and then formalize the community learning problem.

Definition 2.1 (Urban Community). A community consists of (i) a location (i.e., latitude and longitude) of a residential complex and (ii) a neighborhood area (e.g., a circle with radius of 1km).

Usually, in urban areas, a residential complex consists of multiple apartment buildings, where each apartment building has many apartments. In addition, there are many POIs located in the neighborhood area that provide a variety of urban functions and living services to residents in the community. The residents in this community can access these urban facilities and services within a walking distance. Figure 1 shows a sample urban community.

Community detection is a hot topic in social network. There are some social community detection algorithms, such as hierarchical clustering [25], spectral clustering [13], divisive algorithm [46], and modularity-based methods [15, 22, 23]. In the scenario of urban computing, there are some studies that aim to detect urban regions using POI data and human mobility data [12, 59]. In our work, we focus on learning the representation of urban communities. To simplify the spatial data preprocessing, we define an urban community as a circle neighborhood area with a residential complex in the center of the circle area. In the future, we can explore the methods of social community detection with the representation learning framework proposed in our article.

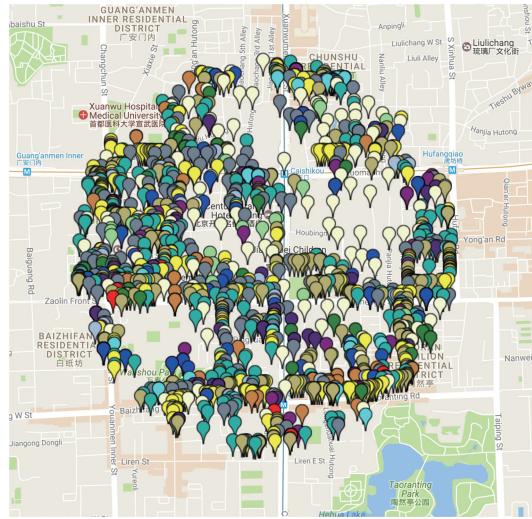


Fig. 1. This is a sample urban community, where the center is a residential complex. The drop pins surrounding the center are POIs located within one kilometer to the residential complex. The different colors of the drop pins represent different POI categories, such as living services, educations, finance, shopping, and restaurants.

Definition 2.2 (Mobility Graph). The mobility graph of a community is a graph extracted from the POIs data and human mobility data of the community. In this graph, POIs are regarded as nodes, and the weights of edges are the human mobility connectivities between two POIs.

In the methodology section, we introduce a probabilistic propagation-based method to compute the human mobility connectivity between two POIs. Figure 2 shows an example of a mobility graph.

Definition 2.3 (Periodic Mobility Graphs). Periodic mobility graphs describe the movements of residents in a community throughout a period of time, which is the aggregation of the daily mobility graphs.

The movements of residents in a community are dynamic and always vary over time. Specifically, human movements are usually periodic [36, 37]. For instance, local residents mostly go to work in the morning and get back home in the afternoon during weekdays. To describe such periodic dynamics of human mobility in a community, we propose to extract the periodic mobility graphs of a community at a daily granularity. In our experiments, we extracted seven periodic mobility graphs (from Monday to Sunday) for each community. In this way, we can learn the representation of a community not only from the structure of such mobility graphs but also the periodic dynamics of such mobility graphs. An example of periodic mobility graphs is shown in Figure 3.

Definition 2.4 (Community Embedding). The embedding of a community is a vector representation of the community. The vector representation describes two types of information about the community: (i) static spatial configuration, i.e., POIs, and (ii) dynamics of human mobility, i.e., the evolving structures of mobility graphs.

It is very important to develop a representation learning model that can take periodic mobility graphs as inputs and output the vector representations of communities.

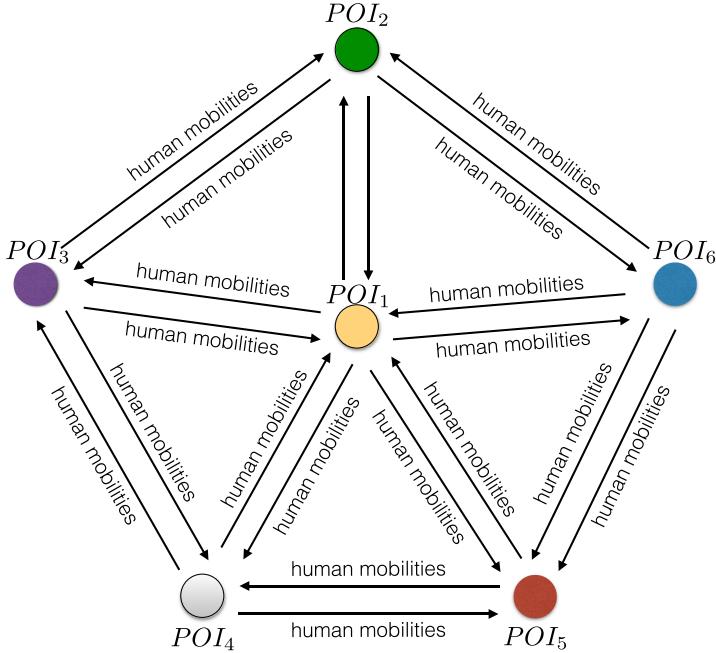


Fig. 2. This is a mobility graph that consists of six POIs. In this graph, each POI is regarded as a node in the mobility graph. The human mobility connectivity between two POIs is regarded as the weight of an edge.

Definition 2.5 (Problem Formulation–Learning Urban Community Structure). By considering the existence of a candidate urban community, in which there are a set of POIs and GPS trajectories of human mobility, we wish to learn the vector representation of the community, such that the learned vector representation can describe not only static spatial configurations, such as POIs and corresponding geographical allocations, but also the dynamic human mobility connectivity of POIs in the community. We formulate this problem as a task of spatial representation learning. Formally, given a set of spatial graphs $G^{(k)} = \{G_1^{(k)}, G_2^{(k)}, \dots, G_M^{(k)}\}$ that describe both POIs and human mobility connectivity between each POI pair for a community c , the spatial representation learning problem aims at learning a mapping function $f(c_k) : G^{(k)} \rightarrow \mathbb{R}^d$ that can map the structural information of multiple mobility graphs into a vector representation for the community c_k . Essentially, there are three major steps: (1) Construct the periodic mobility graph set for a community, (2) collectively learn the POI embedding from multiple mobility graphs, and (3) aggregate and align POI embedding into community embedding.

3 METHODOLOGY

We first present an overview of our proposed framework and then detail the three critical steps: (i) constructing periodic mobility graphs, (ii) collective POI embedding, and (iii) aligning and aggregating POI embeddings into community embeddings.

3.1 Framework Overview

The focus of this article is to develop a representation learning framework of community structure that can captures dynamic changes of community structure, due to the human mobility. Figure 4 shows our proposed framework that consists of three main steps: (i) periodic mobility graph

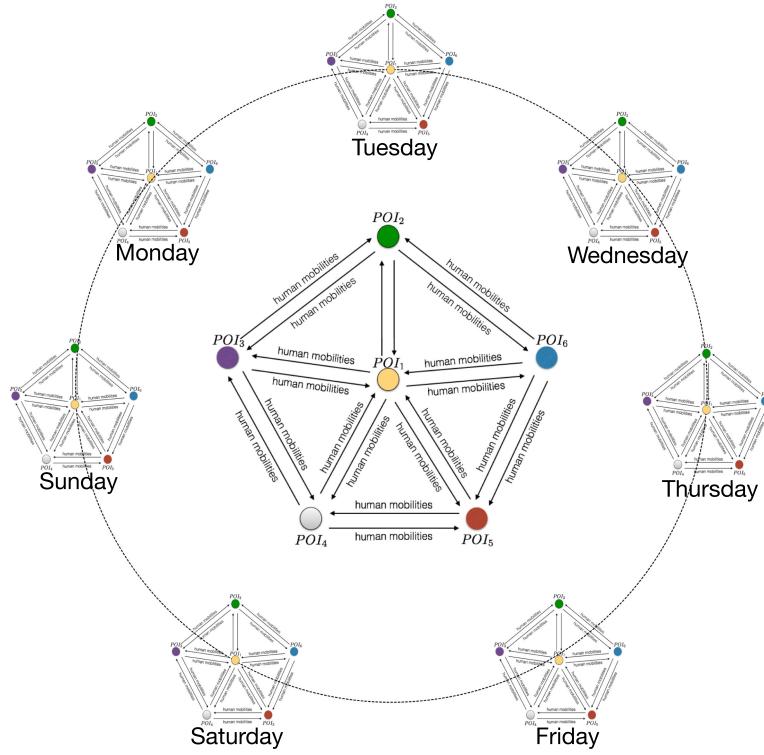


Fig. 3. This is an example of seven periodic mobility graphs, each of which represents the mobility connectivity of the POI graphs on Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday, respectively.

construction, (ii) collective POI embeddings, and (iii) aligning and aggregating POI embeddings into community embeddings. Specifically, in the first step, we construct seven periodic mobility graphs, where vertexes are POIs and edges represent human movement between POIs. Second, we propose a collective deep auto-encoder to learn POI embeddings from the periodic mobility graphs of each community. Finally, we exploit a graph-based unsupervised weighted aggregation method to semantically align and aggregate POI embeddings into the embeddings of POI categories. Then, we further aggregate POI-category embeddings into community embeddings.

3.2 Periodic Mobility Graph Construction

According to Definition 2.3, we aim to learn the representation of the structure of an urban community from mobility graphs that describe seven days of human movements among POIs. To construct periodic mobility graphs, the key challenge is how to extract the connectivity measurements between the POIs of a community from the large-scale human movement data.

Intuitively, people's outdoor activities include the transitions from one POI to another POI, and, ultimately, form massive mobility flows in a community. As a result, human mobility can indicate the connectivity among POIs. Therefore, we estimate the possibility of mobile users that move from one POI to another POI to quantify the mobility connectivity between two POIs. This step is important, because it enables us to measure the human mobility connectivity between two POIs. In this way, we can construct mobility graphs over different days to represent an urban community. The extracted graphs will be fed into the collective embedding model to learn the representation of

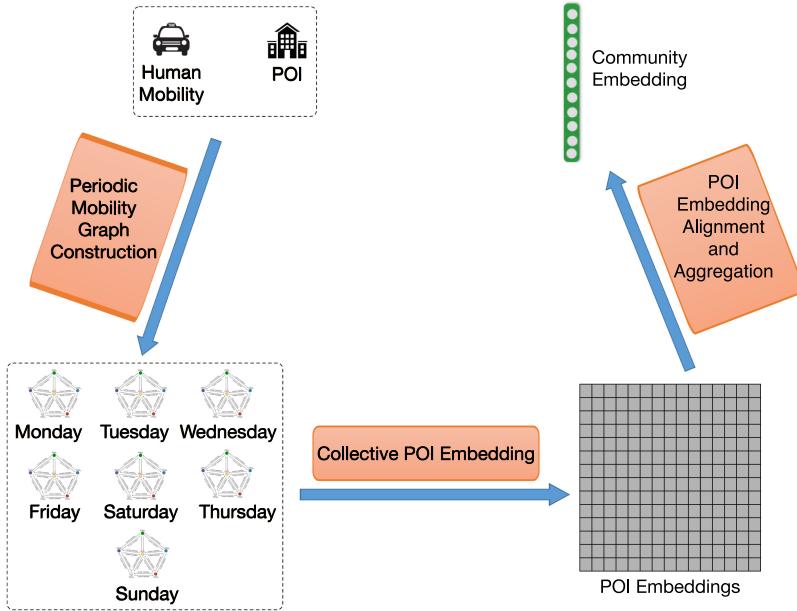


Fig. 4. The overview of the proposed analytical framework.

urban communities. A straightforward method for mobility connectivity estimation between two POIs, e.g., POI(O) and POI(D), is to directly count the total number of visits from POI(O) for POI(D). However, such a simple method highly depends on the availability of high-quality mobility data, in which each trip must include an origin POI and a destination POI. In reality, human mobility data are collected from different devices and sources (e.g., smartphones, cellular stations, GPS-equipped vehicles, location-based services). Therefore, not every trip in the human mobility data includes an accurate origin POI and an accurate destination POI. For instance, a taxi ride usually includes an origin GPS point and a destination GPS point. But the original GPS point is not the origin POI; the destination GPS point is not the destination POI as well. Thus, to develop a generalized estimation method that fits various trajectory data and does not require exact origin POIs and destination POIs, we propose to exploit a probability propagation-based method in Reference [19] and propose the following three-step algorithm:

- Step 1: Propagate visit probability. Given the drop-off point d of a taxi trace, we model the probability of a POI p visited by a passenger as a parametric function, whose input x is the distance between the drop-off point d and the destination POI p :

$$P(x) = \frac{\beta_1}{\beta_2} \cdot x \cdot \exp\left(1 - \frac{x}{\beta_2}\right), \quad (1)$$

where β_1 and β_2 are two given hyper-parameters that control the shape of the function $P(x)$. Figure 5 shows the function graph of $P(x)$ with the maximum visiting probability, $\beta_1 = 0.8$, and the most comfortable walking distance between the drop-off point and the destination POI, $\beta_2 = 100$. We adopt such function to estimate visiting probability from drop-off points to destination POIs, because the function has several mathematical properties as follows:

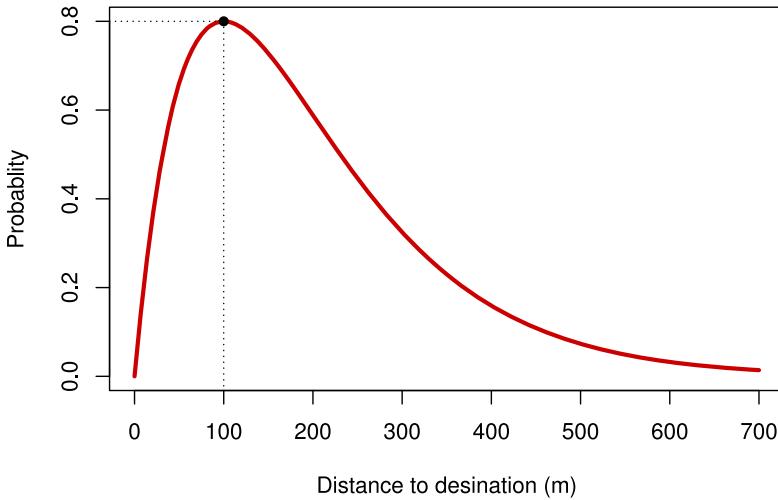


Fig. 5. Probability distribution w.r.t $\beta_1 = 0.8, \beta_2 = 100$.

- $\beta_1 = \max_x P(x)$: β_1 is the maximum value of $P(x)$, and thus β_1 can be explained as how likely a mobile user will visit the destination POI starting from the drop-off point under the function $P(x)$. This mathematical property can allow us to easily and empirically set the maximum probability of visiting the destination POI from the drop-off point.
- $\beta_2 = \arg \max_x P(x)$: β_2 is the value of x when $P(x) = \beta_1$. Since x is the distance between the drop-off point and the destination POI, β_2 can be explained as the most comfortable walking distance between the drop-off point and the destination POI for taxi passengers.
- When $x = 0, P(x) = 0$: Since a taxi may not send passengers into a POI building directly, the drop-off point is usually not the destination POI. A passenger often walks a short distance to reach the destination.
- When $x > \beta_2$, the value $P(x)$ keeps dropping and shows an exponential heavy tail effect. In the problem of visiting probability estimation, the drop-off point is usually close to the destination POI. It is impossible for a mobile user to request a taxi driver to drop off him/her at a place that is very far away from the destination. Hence, when the distance exceeds the most comfortable walking distance β_2 , the probability keeps decreasing.
- When $x < \beta_2$, the value $P(x)$ keeps increasing with distance increasing. The intuition behind this is that the drop-off point is usually not the same as the destination, and there will be a short distance between the drop-off point and the destination. Subjected to the road network, when the destinations locate in the neighboring buildings or plazas, the drop-off points are usually the same, like at the intersections or beside the pavement near buildings. For example, in one commercial circle, restaurants and movie theaters are usually close to each other. When it is dinner time, even if the welcoming theater is closer to the drop-off point than the restaurant, the passengers will be more likely to visit the restaurant instead of the movie theater. Therefore, the rule of “the closer, the more likely to visit” is not applicable to the scenario of the visiting probability. However, if we apply the exponential decay function directly, the visiting probability will decrease with the distance increasing. This means that the visiting probability to the closest POI is the highest. It is inconsistent with the fact.

With this function, we can propagate the visit probability of a passenger from the drop-off point to its surrounding POIs.

- Step 2: Calculate POI visit probability. To evaluate the probability of the i th POI p_i visited by users, we need to aggregate all probabilities from all drop-off points in taxi traces: $\tau(p_i) = \sum_{d \in \mathcal{D}} P(dis(d, p_i))$, where \mathcal{D} is the drop-off point set of taxi traces in the community.
- Step 3: Calculate mobility connectivity between POI pairs. We multiply the $\tau(p_i)$ with $\tau(p_j)$ to describe the possibility of users visiting p_j from p_i . This possibility is used to quantify the mobility connectivity between p_j and p_i . The calculation can be formulated as:

$$\tau_{ij} = \begin{cases} \tau(p_i) \cdot \tau(p_j), & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}. \quad (2)$$

Given a community c_k , we segment POI data and human mobility data into seven parts based on days in a week. Then, we apply the three-step mobility graph construction method over the seven parts of data. After that, we can obtain a graph set $G^{(k)}$ of seven mobility connectivity graphs across POIs, where $G^{(k)} = \{\mathcal{G}_1^{(k)}, \mathcal{G}_2^{(k)}, \dots, \mathcal{G}_t^{(k)}, \dots, \mathcal{G}_7^{(k)}\}$, $\mathcal{G}_t^{(k)}$ denotes the mobility connectivity graph across POIs of the community c_k on the t th day of a week.

3.3 Collective POI Embedding

Since POIs are links between communities and people, mobility-based POI-level features can reveal more patterns about the structure of the community. Along this line, we propose a collective POI embedding method over the periodic mobility graphs, based on auto-encoder.

Before introducing the details of our proposed collective POI embedding method, we first give a brief review of the traditional auto-encoder model. Auto-encoder is an unsupervised neural network model, which projects the instances (in original feature representations) into a lower-dimensional feature space via a series of non-linear mappings. However, the traditional auto-encoder can only take one input in each training iteration. However, since we have constructed periodic mobility graphs to capture the spatiotemporal dynamics of the community structure, we need a collective learning method to learn the embeddings of community structure from the inter-correlations of multiple mobility graphs.

To solve the problem, we propose a collective POI embedding method. Formally, for a given community c_k , the i th row of the constructed periodic graph $\mathcal{G}_t^{(k)}$ is used to represent the i th POI $p_{i,t}^{(k)}$ on the day t of a week. Then, given a POI $p_i^{(k)}$, we have seven vectors $\{\mathbf{p}_{i,t}^{(k)}\}^*$, where $t = 1, 2, \dots, 7$. We utilize these seven vectors as inputs. Meanwhile, we denote the embeddings of the POI p_i on the day t as $\{y_{i,t}^{(k),1}, y_{i,t}^{(k),2}, \dots, y_{i,t}^{(k),o}\}$, at hidden layers $1, 2, \dots, o$ in the encoding step, respectively. The encoding result of p_i in the targeted lower-dimensional feature space can be represented as $\mathbf{z}_i^{(k)} \in \mathbb{R}^N$. To handle the problem of multiple inputs, we add embeddings ensembling process on the $o+1$ layer before generating $\mathbf{z}_i^{(k)}$, as shown in Figure 6. Then, the encoding step can be formulated as:

$$\begin{cases} y_{i,t}^{(k),1} &= \sigma(\mathbf{W}_{i,t}^{(k),1} \mathbf{p}_{i,t}^{(k)} + \mathbf{b}_{i,t}^{(k),1}), \forall t \in \{1, 2, \dots, 7\}, \\ y_{i,t}^{(k),r} &= \sigma(\mathbf{W}_{i,t}^{(k),r} \mathbf{p}_{i,t}^{(k)} + \mathbf{b}_{i,t}^{(k),r}), \forall r \in \{2, 3, \dots, o\}, \\ y_i^{(k),o+1} &= \sigma(\sum_t \mathbf{W}_t^{(k),o+1} y_{i,t}^{(k),o} + \mathbf{b}_t^{(k),o+1}), \\ \mathbf{z}_i^{(k)} &= \sigma(\mathbf{W}^{(k),o+2} y_i^{(k),o+1} + \mathbf{b}^{(k),o+2}), \end{cases} \quad (3)$$

where \mathbf{W} s and \mathbf{b} s denote the weight terms and bias terms, respectively. In particular, $\mathbf{W}_t^{(k)}$ denotes the weight term and $\mathbf{b}_t^{(k)}$ denotes the bias term for $\mathbf{p}_i^{(k)}$ at the embeddings ensembling process.

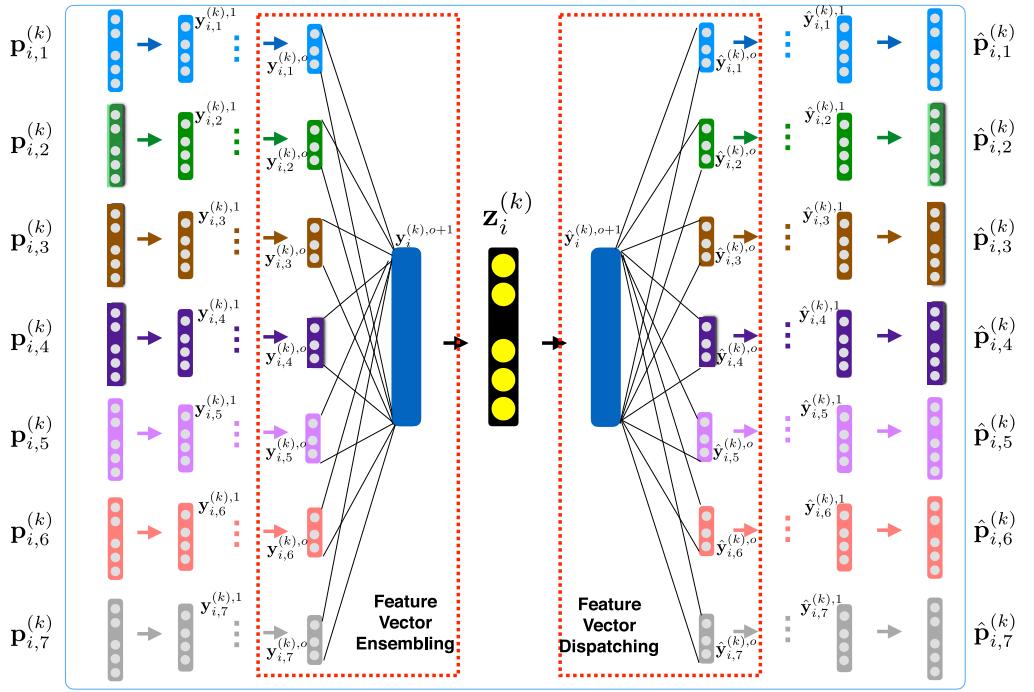


Fig. 6. Collective POI embedding.

In the decoding step, the input will be the embedding $\mathbf{z}_i^{(k)}$ (the output of the encoding step), and the final output will be the reconstructed vector $\hat{\mathbf{p}}_{i,t}^{(k)}$. First, we dispatch $\mathbf{z}_i^{(k)}$ into seven latent vectors for each day. Then, the reconstructed embeddings at each hidden layer can be represented as $\hat{\mathbf{y}}_{i,t}^{(k,o)}, \hat{\mathbf{y}}_{i,t}^{(k,o-1)}, \dots, \hat{\mathbf{y}}_{i,t}^{(k,1)}$. The relationship among these vector variables can be denoted as

$$\begin{cases} \hat{\mathbf{y}}_i^{(k,o+1)} = \sigma(\hat{\mathbf{W}}^{(k),o+2}\mathbf{z}_i^{(k)} + \hat{\mathbf{b}}^{(k),o+2}), \\ \hat{\mathbf{y}}_{i,t}^{(k,o)} = \sigma(\hat{\mathbf{W}}_t^{(k),o+1}\hat{\mathbf{y}}_i^{(k),o+1} + \hat{\mathbf{b}}_t^{(k),o+1}), \\ \hat{\mathbf{y}}_{i,t}^{(k,r-1)} = \sigma(\hat{\mathbf{W}}_{i,t}^{(k),r}\hat{\mathbf{y}}_{i,t}^{(k),r} + \hat{\mathbf{b}}_{i,t}^{(k),r}), \forall r \in \{2, 3, \dots, o\}, \\ \hat{\mathbf{p}}_{i,t}^{(k)} = \sigma(\hat{\mathbf{W}}_{i,t}^{(k),1}\hat{\mathbf{y}}_{i,t}^{(k,1)} + \hat{\mathbf{b}}_{i,t}^{(k,1)}), \end{cases} \quad (4)$$

where \mathbf{W} s and \mathbf{b} s denote the weight terms and bias terms, respectively. In particular, $\hat{\mathbf{W}}_t^{(k)}$ denotes the weight term and $\hat{\mathbf{b}}_t^{(k)}$ denotes the bias term for $\mathbf{p}_i^{(k)}$ at the embeddings dispatching process.

For the loss function, to tackle the sparsity problem (there are 0 for both $\mathbf{z}_i^{(k)}$ and $\hat{\mathbf{p}}_{i,t}^{(k)}$), we assign a larger weight for the loss introduced by the non-zero features. Then, we aggregate the loss of each day t to obtain the final loss function:

$$\mathcal{L}^{(k)} = \sum_{t \in \{1, 2, \dots, 7\}} \sum_i \|(\mathbf{p}_{i,t}^{(k)} - \hat{\mathbf{p}}_{i,t}^{(k)}) \odot \mathbf{v}_{i,t}^{(k)}\|_2^2, \quad (5)$$

where $\mathbf{v}_{i,t}^{(k)}$ is the weight vector corresponding to the input $\mathbf{p}_{i,t}^{(k)}$. The entries in $\mathbf{v}_{i,t}^{(k)}$ corresponding to non-zero elements are set to a value λ ($\lambda > 1$ denotes a larger weight to fit these features); the rest entries in $\mathbf{v}_{i,t}^{(k)}$ are set to 1.

3.4 Aligning and Aggregating POI Embeddings to Community Embeddings

With the collective deep auto-encoder, we can obtain the embeddings (vector representations) of POIs. However, we aim to extract the embeddings of urban communities. Since an urban community has many POIs, our objective is to weight, align, and aggregate the POI embeddings into the vector representations of urban communities.

To achieve this goal, we proceed with two steps: (1) aggregating POI embeddings to POI-category embeddings and (2) aggregating POI-category embeddings into community embeddings.

(1) *Aggregating POI embeddings to POI-category embeddings*: Be sure to notice that different urban communities might have different numbers of POIs, and, thus, the sizes of the mobility graphs vary over communities. It therefore is very challenging to semantically align and aggregate the embeddings of individual POIs into the embeddings of a community. Unlike individual POIs, POI categories can be semantically aligned. More importantly, different from the number of POIs in urban communities, the number of POI categories is fixed in every urban community. Along this line, we propose to aggregate POI embeddings into the embeddings of POI categories for each community. Intuitively, given a POI category, for example, education, we can sum up the embeddings of all the education POIs into the POI-category embedding of education. However, such simple summation ignores the fact that the latent features in a POI embedding indeed have different importances and weights. To quantify the importance of each latent feature in the POI embeddings, we develop an unsupervised graph-based weighted aggregation approach.

Traditionally, if the learned latent representations of urban communities are to be applied to a specific application problem, e.g., crime rate prediction, then a straightforward method for estimating feature weights is to directly calculate the statistical relevance between the latent features of the learned embedding vectors and the crime rates. However, this idea is not generalized and highly depends on the availability of the target values in prediction tasks. Our proposed unsupervised weighting method is important, because the unsupervised fashion is designed to ensure that the learning of feature weights does not depend on the availability of the target values to be predicted and thus is independent from the application problems to be applied.

Formally, given a community c_k , let $\tilde{\mathcal{G}}^{(k)}$ be the embedding vectors of all the POIs in the community c_k . $\tilde{\mathcal{G}}^{(k)} \in \mathbb{R}^{M \times N}$, where M is the number of POIs and N is the number of latent features in the POI embedding vectors, each row of $\tilde{\mathcal{G}}^{(k)}$ is the embedding (latent feature vector) of a POI, each column of $\tilde{\mathcal{G}}^{(k)}$ is a latent feature (i.e., a dimension of the latent feature space) in POI embeddings.

The general idea of our proposed graph-based weighting method is as follows: We first create a POI similarity graph, where a vertex is a POI, and the weight of an edge is the similarity of the two embedding vectors of two corresponding POIs. Then, we create an importance measurement to quantify the importance of a latent feature in the embedding feature space by exploiting such a POI similarity graph. Specifically, for two POIs that are highly similar (a high-weight edge in the graph), if the targeted latent feature is important, then the two values of the latent feature in the two POI embedding vectors should be consistently similar, and we will increase its importance. Otherwise, we will penalize its importance measurement. Likewise, for two POIs that are not similar (a low-weight edge in the graph), if the targeted latent feature is important, then the two values of the latent feature in the two POI embedding vectors should be different.

Given the l th feature in the POI embedding vectors, we evaluate its importance based on the following steps:

- First, for every POI pair, we calculate the similarity between two POIs based on their embedding vectors using the cosine similarity, which is given by

$$\text{sim}_{i,j} = \frac{\sum_l \tilde{\mathcal{G}}^{(k)}[i,l] \times \tilde{\mathcal{G}}^{(k)}[j,l]}{\sqrt{\sum_l \tilde{\mathcal{G}}^{(k)}[i,l]^2} \times \sqrt{\sum_l \tilde{\mathcal{G}}^{(k)}[j,l]^2}}. \quad (6)$$

Based on the above, we can build a POI similarity graph.

- Second, we calculate the weight of l th dimension of the feature by examining every edge (each POI pair) of the POI similarity graph. The weight is given by

$$w_l^{(k)} = \frac{\sum_{i \in c_k} \sum_{j \in c_k} \text{sim}_{i,j} \times |\tilde{\mathcal{G}}^{(k)}[i,l] - \tilde{\mathcal{G}}^{(k)}[j,l]|}{M}. \quad (7)$$

The intuition behind Equation (6) and Equation (7) is as follows: If the l th dimension of the latent feature makes more sense, when POI p_i and p_j are very similar, then the difference of p_i and p_j on the l th dimension ($|\tilde{\mathcal{G}}^{(k)}[i,l] - \tilde{\mathcal{G}}^{(k)}[j,l]|$) should be very small. Therefore, if the l th dimension of the latent feature does not make much sense, then $|g[i,l] - g[j,l]|$ will increase; if p_i and p_j are very similar, then $\text{sim}_{i,j}$ will further penalize $|g[i,l] - g[j,l]|$.

By using this method, we can obtain the latent feature weight set $\mathbf{w}^{(k)} = \{w_1^{(k)}, w_2^{(k)}, \dots, w_l^{(k)}, \dots, w_N^{(k)}\}$.

$$\hat{\mathcal{G}}^{(k)}[s,l] = \sum_{p_i \in \Phi_s} \tilde{\mathcal{G}}^{(k)}[i,l] \times w_i^{(k)}, \quad (8)$$

where $\hat{\mathcal{G}}^{(k)}$ is the POI-category embedding graph for the community c_k and Φ_s is the s th POI category.

(2) *POI alignment*: Given a community c_k , we align each row of $\hat{\mathcal{G}}^{(k)}$ into a vector: $\mathbf{G}^{(K)} = (\hat{\mathcal{G}}^{(k)}[1,*], \hat{\mathcal{G}}^{(k)}[2,*], \dots, \hat{\mathcal{G}}^{(k)}[s,*])^T$, where $\mathbf{G}^{(K)}$ is the aligned community embedding that is also the output of the proposed representation learning framework.

4 APPLICATIONS

To evaluate and interpret the embeddings of residential communities, we apply our proposed embedding framework to two applications: (1) predicting willing to pay (WTP) for urban communities and (2) spotting vibrant urban communities.

4.1 Predicting Willing to Pay (WTP)

Empirical studies have shown that the WTP for communities can be reflected by the return rates of real estate prices over a market period, i.e., rising or falling markets [14, 21]. Therefore, given a market period, WTP can be measured by the ratio of the price increase relative to the starting price of a market period, i.e., $r = \frac{P_f - P_i}{P_i}$, where P_f and P_i denote the final and initial prices, respectively.

In this application, we first learn and extract the representation features of urban communities using the proposed collective embedding method. Then, we calculate the benchmark WTP values for each community. Finally, we utilize linear regression to predict the WTP for each community.

4.2 Spotting Vibrant Urban Communities

We aim to spot vibrant urban communities. Intuitively, a community can be considered prosperous and vibrant if the community can attract a great number of mobile users to visit and consume or if the community can provide a variety of products and services to residents. Therefore, we propose a measurement, which we call *community vibrancy* for simplicity, to measure the combined effect of both the density and diversity of consumption check-in activities. Specifically, for the community c_k , we first count the number of consumption check-in events as the density of consumption activities, denoted by $\text{freq}^{(k)}$. In addition, we measure the diversity by calculating

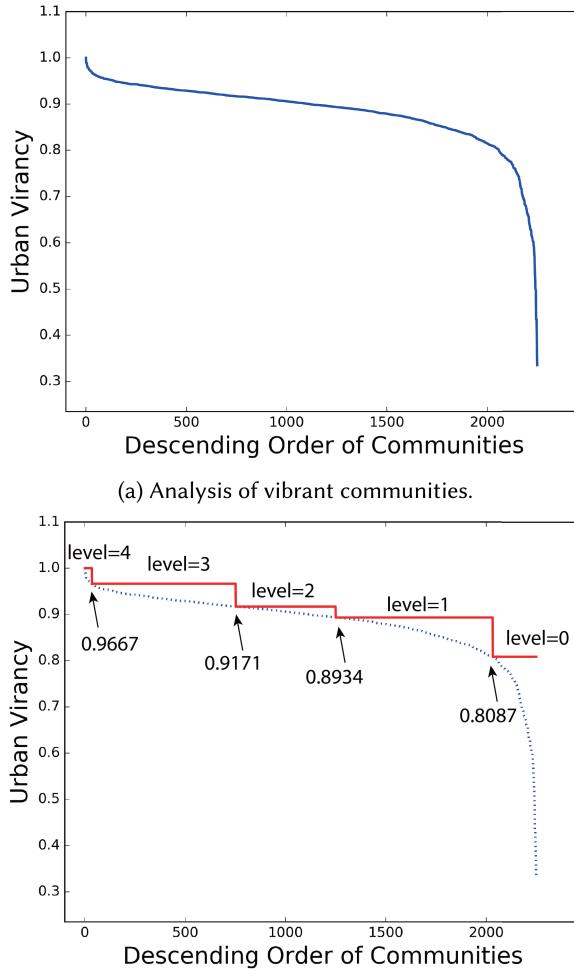


Fig. 7. Analysis of community vibrancy.

the entropy of check-in events over different POI categories: $div^{(k)} = \sum_{i=1}^m freq_i^{(k)}$, where $freq_i^{(k)}$ denotes the consumption activity amount of the i th POI category in the community c_k . Finally, we fuse the density and diversity using the harmonic mean to represent the score of community vibrancy: $u_k = \frac{2 \times freq^{(k)} \times div^{(k)}}{freq^{(k)} + div^{(k)}}$.

Figure 7(a) shows that all the communities are sorted in a descending order in terms of the computed vibrancy scores. From the curve in Figure 7(b), we can identify four inflection points, representing the vibrancy scores: 0.9667, 0.9171, 0.8934, and 0.8087, respectively. The four inflection points are used to segment the curve into five segments. Accordingly, we can assign five-level ratings to each segment as its ranking relevance label, ranging from 0 to 4. We observe that the distribution of the community vibrancy scores complies with a power-law distribution, which means that only a small number of residential communities are highly vibrant while most communities are around the mean value of the vibrancy scores. This observation is consistent with our common sense that most people are middle-class and only a small group of people are rich.

Table 1. Statistics of the Experimental Data

Data Sources	Properties	Statistics
Taxi Traces	Number of taxis	13,597
	Effective days	92
	Time period	Apr. – Aug. 2012
	Number of trips	8,202,012
	Number of GPS points	111,602
	Total distance(km)	61,269,029
Residential Communities	Number of residential communities	2,990
	Latitude and Longitude	
	Time period of transactions	04/2011 – 09/2012
POIs	Number of POIs	328668
	Number of POI categories	20
	Latitude and Longitude	
Check-Ins	Number of check-in events	2,762,128
	Number of POI categories	20
	Time Period	01/2012–12/2012

Formally, let u_k be the vibrancy score of the community c_k , and $rank^{(k)}$ denote the ranking of the community c_k based on the vibrancy score u_k . Then, the problem of ranking vibrant communities can be formulated as follows: Given the POI and human mobility data, we aim to predict the vibrancy ranking $rank^{(k)}$ of the community c_k using the community embeddings learned by the our proposed framework.

5 EXPERIMENTAL RESULTS

We provide an empirical evaluation of the performances of the proposed method on real-world urban community and human mobility data.

5.1 Data Description

Table 1 shows the statistics of four data sources used in the experiment. The taxi GPS traces are collected from a Beijing taxi company. Each trajectory contains trip ID, distance(m), travel time(s), average speed(km/h), pick-up time and drop-off time, pick-up point and drop-off point. We also extracted POIs related data from www.dianping.com, which is a business review site in China. In addition, we obtain the Beijing residential community data by crawling www.soufun.com, which is the largest real-estate online system in China; and we obtain the check-in data of Beijing by crawling www.jiepang.com, which is a Chinese version of Foursquare. Each check-in event includes POI name, POI category, address, longitude, and latitude.

5.2 The Application of WTP Prediction

5.2.1 Experimental Setup.

- (1) Baselines. To evaluate the effectiveness our proposed collective embedding method, we compare six feature sets:
 - Explicit Features (EF): Specifically, the explicit features are explicitly defined and extracted from the data as follows: (i) *POI numbers per category*: There are 20 POI categories including vehicle service, car dealer, repair & maintenance, motorbike dealer & service, food & beverage, shopping, daily life service, sports recreation, medical

Table 2. The Performance Comparison on WTP Prediction

Feature set	ELF	LF	EF	V-1	V-2	V-3
RMSE	0.0036	0.0057	0.0422	0.0273	0.0350	0.0193

service, lodging, tourist, real estate, government & non-government, organization, culture & education, transportation, finance & insurance, company & factory, road furniture, named place & address, public service; (ii) *Average commute distance*; (iii) *Average commute speed*; (iv) *Average commute time*; (v) *Number of mobilities*; (vi) *Average distance between POIs*.

- Latent Features (**LF**): Specifically, the latent features are learned from the proposed collective embedding method.
- The combination of EF and LF (**ELF**). Specifically, we combine both the explicit features via traditional feature extraction and the latent features via representation learning together into a new feature set.
- Variation of step1 (**V-1**). In the first step of the learning framework, we propose a probabilistic way to derive the mobility graphs over POIs for different days. There is a simple variation that use distance-based matching of the records in the trajectories with the POIs and build a transition graph deterministically. We modify the first step and keep other parts of the learning framework the same. We use this version of framework to generate features.
- Variation of step2 (**V-2**). In the second step of the learning framework, we propose a collective learned method based on Autoencoder. An alternative way is to, first, derive different embeddings using different graphs and then compute the POI embedding as an average of the embeddings. We modify the second step and keep other parts of the learning framework the same. We use this version of framework to generate features.
- Variation of step3 (**V-3**). In the third step of the learning framework, we propose graph-based method to aggregate the POI embeddings into community embeddings. An alternative way is just averaging over the POI embeddings in the community to derive the community embedding. We modify the third step and keep other parts of the learning framework the same. We use this version of framework to generate features.

(2) Evaluation Metrics.

We utilize the root-mean-square error (RMSE) to evaluate the performance.

5.2.2 *Results and Analysis.* Table 2 shows the performance comparison of six feature sets in term of RMSE. In all cases, we observe that the combination of explicit features and latent features achieves the best performance, while the explicit feature has the highest errors. For the three variations of our proposed learning framework, the performances are worse than the latent features. This validates the necessity of designing such three steps for learning.

5.3 Spotting Vibrant Urban Communities

5.3.1 *Performance Comparison with Application Related Methods.* As mentioned before, we apply the proposed collective embedding method to ranking high-rated urban communities as an application. Here, we choose some application related (ranking) methods for comparison to demonstrate the effectiveness of our proposed method.

(1) Baseline Algorithms.

To show the effectiveness of the collective embedding framework, we compare the performances of different combinations of feature sets and ranking algorithms.

First, we used five learning methods to rank (LTR) algorithms for comparison:

- Multiple Additive Regression Trees (**MART**) [17]: This is a boosted tree model in which the output of the model is a linear combination of the outputs of a set of regression trees. MART is a class of boosting algorithms that may be viewed as performing gradient descent in function space, using regression trees.
- RankBoost (**RB**) [16]: This is a boosted pairwise ranking method that trains multiple weak rankers and combines their outputs as final ranking. The basic idea of RankBoost is to formalize learning to rank as a problem of binary classification on instance pairs and then to adopt boosting approach. Like all boosting algorithms, RankBoost trains one weak ranker at each round of iteration and combines these weak rankers as the final ranking function. After each round, the document pairs are re-weighted: It decreases the weight of correctly ranked pairs and increases the weight of wrongly ranked pairs .
- LambdaMART (**LM**) [9]: This is the boosted tree version of LambdaRank, which is based on RankNet. LambdaMART combines MART and LambdaRank.
- ListNet (**LN**) [10]: This is a listwise ranking model with permutation top- k ranking likelihood as objective function. It introduces two probability models, respectively referred to as permutation probability and top- k probability, to define a listwise loss function for learning. Neural Network and Gradient Descent are then employed as model and algorithm in the learning method.
- RankNet (**RN**) [8]: This uses a neural network to model the underlying probabilistic cost function.

In addition, we utilize six feature sets mentioned in Section 5.2.1 for comparison.

Finally, we create 30 combinations of features and rankers for comparisons. We use “-” between a feature set and a ranker to denote a combination, for instance, “ELF-MART.” We utilize RTree¹ to index geographic items (i.e., taxi and bus trajectories, checkins, etc.) and extract the defined features. For these five LTR algorithms, we use RankLib.² We set the number of trees = 500, the number of leaves = 10, the number of threshold candidates = 256, and the learning rate = 0.1 for MART and LambdaMART. We set the number of iteration = 300, the number of threshold candidates = 10 for RankBoost. We set learning rate = 0.0005, number of hidden layers = 1, the number of hidden nodes per layer = 10, and the number of epochs to train for ListNet and RankNet both. After we generate the data pairs $\{feature, ranking\ relevance\}$, we shuffle the data pairs and select 80% for training and 20% for testing, where *feature* refers to EF, LF, or ELF and *ranking relevance* is based on corresponding vibrancy values.

(2) Evaluation Metrics.

Normalized Discounted Cumulative Gain (NDCG@N). The discounted cumulative gain (DCG@N) is given by $DCG[n] = \begin{cases} rel_n & if n = 1 \\ DCG[n - 1] + \frac{rel_n}{log_2 n}, & if n >= 2 \end{cases}$, where rel_n denotes the ranking relevance of the n th community, defined in Figure 7(b). Later, given the ideal discounted cumulative gain DCG' , NDCG at the n th position can be computed as $NDCG[n] = \frac{DCG[n]}{DCG'[n]}$. The larger NDCG@N is, the higher top-N ranking accuracy is.

Kendall’s Tau Coefficient. Kendall’s Tau coefficient (or Tau for short) is a measure of rank correlation, i.e., the similarity of the orderings of the data. Let us assume that each community i is associated with a benchmark score y_i and a predicted score f_i . Then, for a community pair $< i, j >$, $< i, j >$ is said to be *concordant* if both $y_i > y_j$ and $f_i > f_j$ or if

¹<https://pypi.python.org/pypi/Rtree/>.

²<http://sourceforge.net/p/lemur/wiki/RankLib/>.

both $y_i < y_j$ and $f_i < f_j$. Also, $\langle i, j \rangle$ is said to be *discordant* if both $y_i < y_j$ and $f_i > f_j$ or if both $y_i < y_j$ and $f_i > f_j$. Tau is given by $\text{Tau} = \frac{\#_{\text{conc}} - \#_{\text{disc}}}{\#_{\text{conc}} + \#_{\text{disc}}}$.

F-measure@N. F-measure@N incorporates both precision and recall in a single metric by taking their harmonic mean: $F@N = \frac{2 \times \text{Precision}@N \times \text{Recall}@N}{\text{Precision}@N + \text{Recall}@N}$. Since we use a five-level rating system ($4 > 3 > 2 > 1 > 0$) instead of binary rating, we treat the rating ≥ 3 as “high-vibrancy” and the rating < 3 as “low-vibrancy.” Given a top-N community list E_N sorted in a descending order of the prediction values, the precision and recall are defined as $\text{Precision}@N = \frac{|E_N \cap E_{\geq 3}|}{N}$ and $\text{Recall}@N = \frac{|E_N \cap E_{\geq 3}|}{|E_{\geq 3}|}$, where $E_{\geq 3}$ are the communities whose ratings are greater or equal to 3.

- (3) Results and Analysis. Figure 8 shows the performance comparison of the 15 combinations of the feature sets and the ranking algorithms in terms of Tau, NDCG@N, and F-measure@N. In all cases, we observe a significant improvement by considering the learned embeddings with respect to baselines.

First, we control the ranker and investigate the effectiveness of different feature sets. Among the five rankers, the combination of explicit features and latent features performs the best. Besides, we observe that the latent features outperform the explicit features. In particular, for NDCG@N, when N is getting larger, the results clearly demonstrate the superiority of the latent features learned by our framework. This observation proves that the latent features are discriminative for spotting top vibrant communities. A potential interpretation of this observation is that human mobility in dynamic spatiotemporal graphs provide more information about community structures than the static geographical locations. When we combine the latent and explicit features together, both the dynamic and static structural information of a community are combined to provide a more comprehensive and effective representation for communities. Therefore, the predictive accuracies are significantly improved.

5.3.2 Comparison with Representation Learning Algorithms. We compare our proposed representation learning framework with other state-of-art representation learning algorithms to evaluate the representation learning performance.

- (1) Baseline Algorithms.

We take three state-of-art representation learning algorithms as baselines, including Skip-gram, Restricted Boltzmann Machines (RBMs), and Non-negative Matrix Factorization (NMF).

- Skip-gram: This is a type of Word2vec model that is used to produce word embeddings. Skip-gram uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words [44].
- RBMs: This is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs. RBMs are a q two-layer undirected graphical model that can produce distributed representations of the input and perform well in terms of retrieval accuracy [29].
- NMF: This is a technique that learns a low-dimensional representation of a dataset. When applying NMF over a matrix, the factorized sub-matrix can be interpreted as the latent representation of the original matrix [49].

We use these three baselines and our proposed framework to generate representation learning results. Then, we feed these representation learning results into five LTR algorithms to examine the representation learning performance. After we generate the data

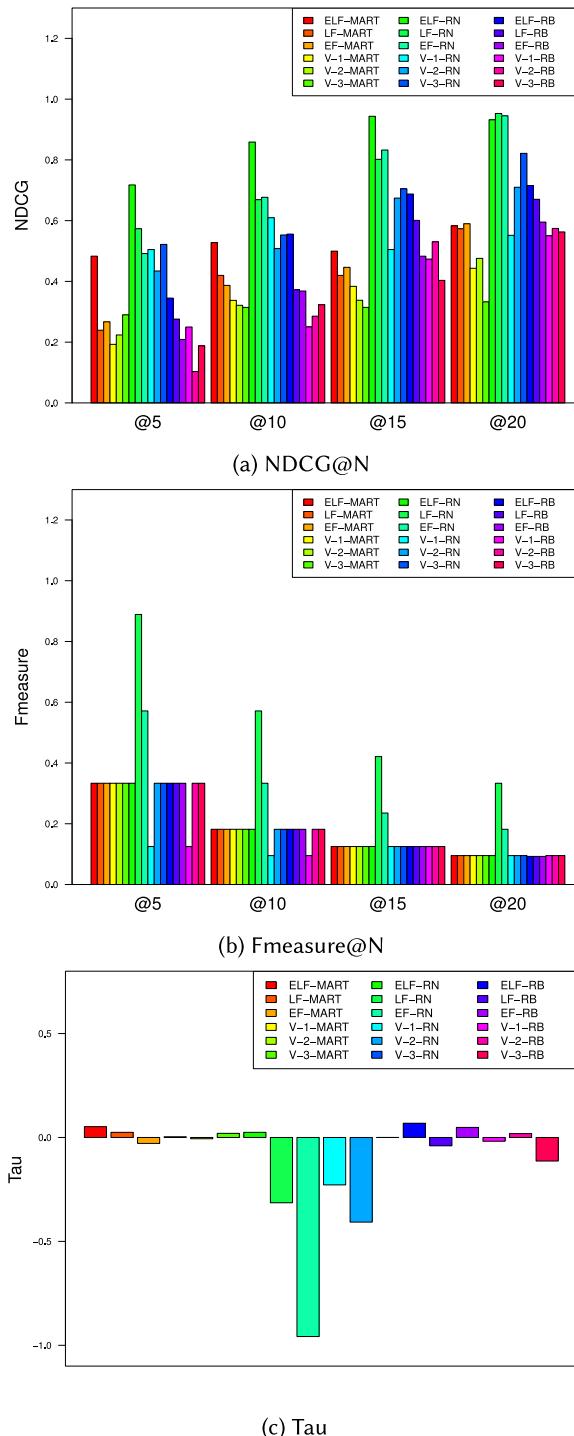


Fig. 8. The overall performance comparisons of the 15 feature and ranker combinations in terms of NDCG, F-measure, and Tau.

pairs $\{feature, ranking\ relevance\}$, we shuffle the data pairs and select 80% for training and 20% for testing, where *feature* refers to EF, LF, or ELF and *ranking relevance* is based on corresponding vibrancy values.

(2) Evaluation Metric.

We compare the performance of our proposed framework, Skip-gram, RBMs, and NMF, in terms of NDCG@N.

(3) Results and Analysis.

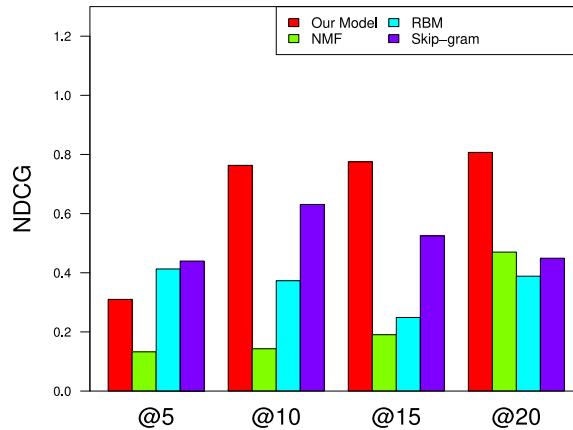
The experimental results are shown in Figure 9. As we can see in Figure 9, in most cases, our proposed model outperforms the skip-gram model. However, for NDCG@5, the skip-gram is better than our model in some cases. A potential explanation is that the skip-gram algorithm weighs nearby context words more heavily than distant context words. In this way, the connectivities of the neighboring POIs in a community play a more important role in the skip-gram algorithm for identifying top vibrant communities. However, the auto-encoder is still better suited for the spatial graph embedding scenario than the skip-gram method. There are two main reasons.

- (a) First, the skip-gram algorithm is originally designed for word embedding to consider the semantics of neighboring words in the sentences. If we regard an urban community as a document and regard a POI as a word, then it is very difficult to define what is “neighboring words in the sentences.” In the spatial scenario, human mobility connectivity is related to but not totally determined by geographic distances. No matter whether the definition of “neighboring in the sentences” is based on distance or human mobility connectivity, the POIs are difficult to be organized as a “semantic” sentence properly. However, if we represent a mix of POIs, human mobility data, and urban communities into graphs, Autoencoder is capable of projecting these graphs into lower-dimensional vectors, while reserving the relationships between POIs in the embedded vector implicitly. This mechanism does not require us to explicitly define what “neighboring words in the sentences” are.
- (b) Second, in this article, we consider the periodical patterns of the community structure. Therefore, we need to collectively model the inputs of different days in a week. Thanks to the nature of the neural network, Autoencoder can be easily modified and improved to meet the requirement than the skip-gram model.

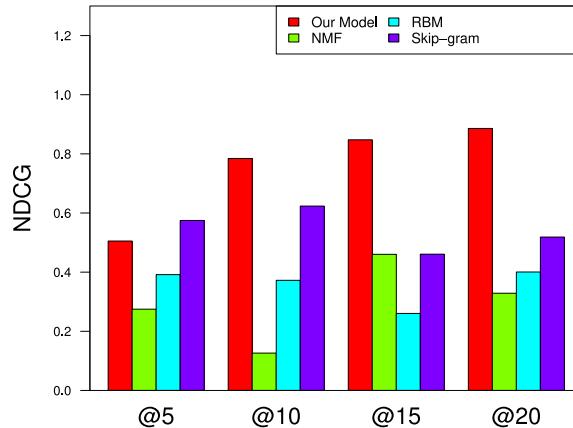
5.4 Robustness Check

We apply the learned embeddings and the ranking algorithms to different subgroups of the communities to examine the robustness of our method in these subgroups. We used two grouping methods to segment the communities into multiple subgroups: (i) neighborhood profile-based grouping and (ii) administrative district-based grouping. For (i), we applied k -means [31] to cluster the communities into five groups. The communities in each group generally share similar functionality and representations. For (ii), for grouping, we chose four administrative districts in Beijing: Haidian, Chaoyang, Xicheng, and Dongcheng, because most of the communities are located the above four districts. Later, we conduct the robustness check from two perspectives.

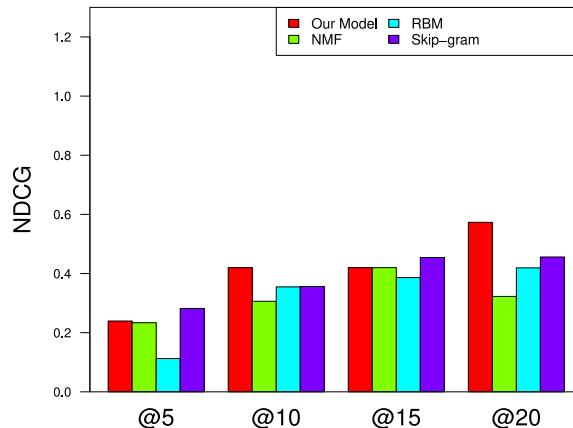
(i) NDCG@N performance comparison. We intend to answer the following question: Compared to the accuracy of our method in all the communities, will our proposed method be consistently effective in the community subgroups? From Figure 8, we have observed that the “ELF-LN” combination performs the best in all the communities. Therefore, we pick “ELF-LN” and examine the effectiveness of “ELF-LN” on the community subgroups. Figure 10 indicates that, for neighborhood profile-based grouping, “ELF-LN” is consistently accurate in the neighborhood profile-based



(a) NDCG@N comparisons over LambdaMART

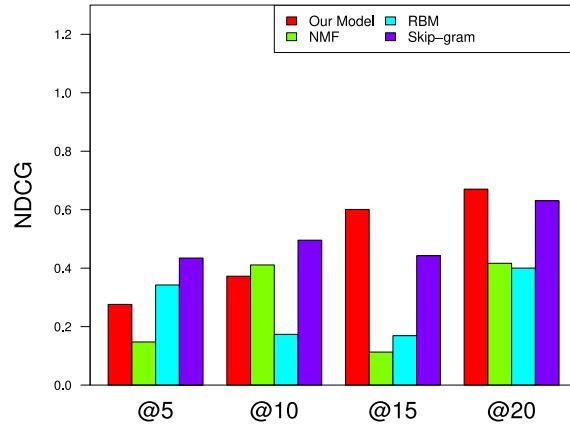


(b) NDCG@N comparisons over ListNet

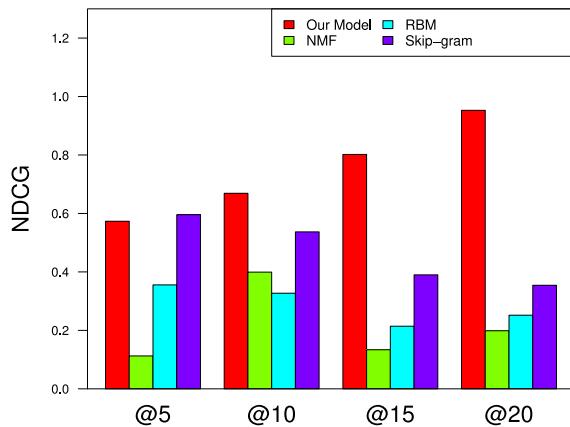


(c) NDCG@N comparisons over MART

Fig. 9. The representation learning comparisons of the Skip-gram, RBMs, NMF, and our proposed framework in terms of NDCG@N.



(d) NDCG@N comparisons over RankBoost



(e) NDCG@N comparisons over RankNet

Fig. 9. Continued.

community subgroups in terms of NDCG@N. We can obtain similar observations in the administrative district-based community subgroups.

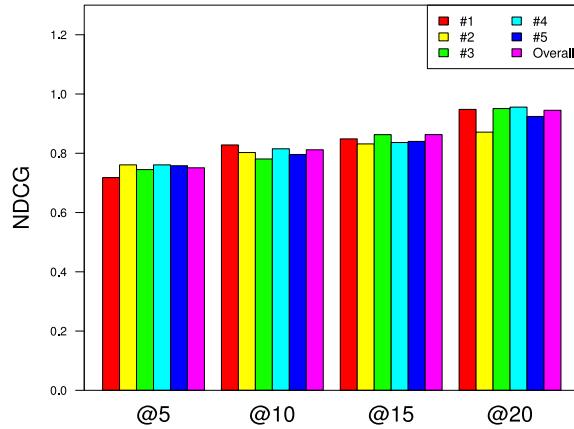
(ii) **Average variance of performance.** For each feature set, we measure the performance variance of all the feature-ranker combinations in terms of NDCG, F-measure, and Tau. Table 3 and Table 4 show that (1) overall, the variance of the “ELF” feature set is the smallest, and (2) the latent features in community embeddings (“LF”) achieve the second smallest variance and are much better than the explicit extracted features.

The results validate the robustness of our method in different community subgroups.

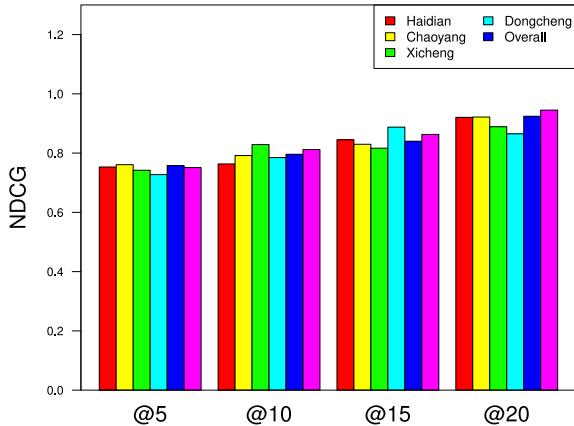
5.5 Investigation of Community Structure Properties

We investigate the structure of urban communities in Beijing in two aspects, (i) community connectivities and (ii) the learned representation of the community structure.

5.5.1 Community Connectivities. We utilize traffic flow as the estimation of the community connectivities. As shown in Figure 11, we use heat map to visualize the results. The darker the color,



(a) The NDCG@Ns of ELF-LN over the five neighborhood profile based subgroups and all the communities.



(b) The NDCG@Ns of ELF-LN over the four administrative district based subgroups and all the communities.

Fig. 10. The NDCG@Ns of the ELF-LN combination over community subgroups. ($N = 5, 10, 15, 20$).

the higher the connectivity. We can observe that communities with high connectivities are mainly distributed around the main road network of Beijing, which demonstrates that the convenience of transportation utilities contributes a lot to connectivities.

5.5.2 The Learned Representation of the Community Structure. For simplicity, we choose two similar communities with similar POI distributions in different POI categories, as shown in Table 5. Then, we utilize heat map to visualize the learned representations of community structures of these two communities. In Figure 12, each column represents the corresponding dimension of the learned representation space, and each row represents the corresponding community. Moreover, the color represents the value of the corresponding dimension. The darker the color, the higher the value. We can observe that for these two communities with similar POI distributions, the learned representations of community structures are still very similar.

Table 3. The Performance Variances of Different Feature Sets over Neighborhood Profile-based Community Subgroups

Features	Variances on NDCG @5	Variances on NDCG @10	Variances on NDCG @15	Variances on NDCG @20	Variances on Fmeasure @5	Variances on Fmeasure @10	Variances on Fmeasure @15	Variances on Fmeasure @20	Tau
ELF	0.0527	0.0581	0.0921	0.0652	0.0019	0.0023	0.0039	0.0059	0.0526
LF	0.0797	0.0825	0.1081	0.1072	0.0018	0.0009	0.0012	0.0019	0.0888
EF	0.1609	0.2116	0.2754	0.2422	0.0022	0.0023	0.0039	0.0059	0.0987

Table 4. The Performance Variances of Different Feature Sets over Administrative District-based Community Subgroups

Features	Variances on NDCG @5	Variances on NDCG @10	Variances on NDCG @15	Variances on NDCG @20	Variances on Fmeasure @5	Variances on Fmeasure @10	Variances on Fmeasure @15	Variances on Fmeasure @20	Tau
ELF	0.0871	0.0867	0.0392	0.0963	0.0019	0.0022	0.0039	0.0059	0.0285
LF	0.0913	0.1103	0.0828	0.1389	0.0022	0.0023	0.0039	0.0059	0.0925
EF	0.2970	0.2937	0.2811	0.1961	0.0019	0.0023	0.0039	0.0059	0.1027

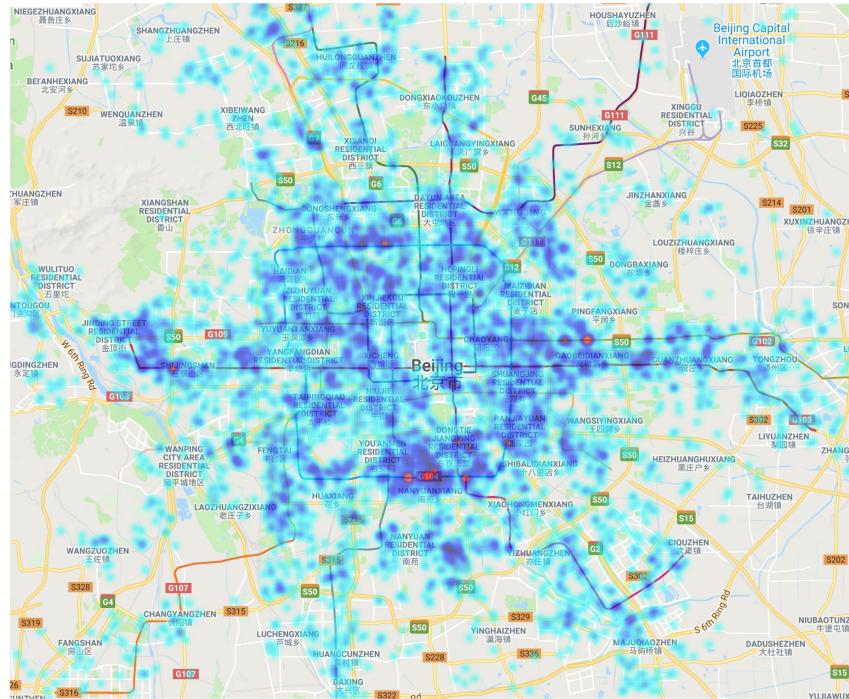


Fig. 11. Visualization of the community connectivities.

Table 5. The POI Distributions of Two Selected Communities

	Restaurant	Business/Public Agency	Entertainment	Transportation/Lodge	Others
Community 1	3,572	7,858	1,951	2,136	311
Community 2	2,395	7,487	1,968	2,423	420

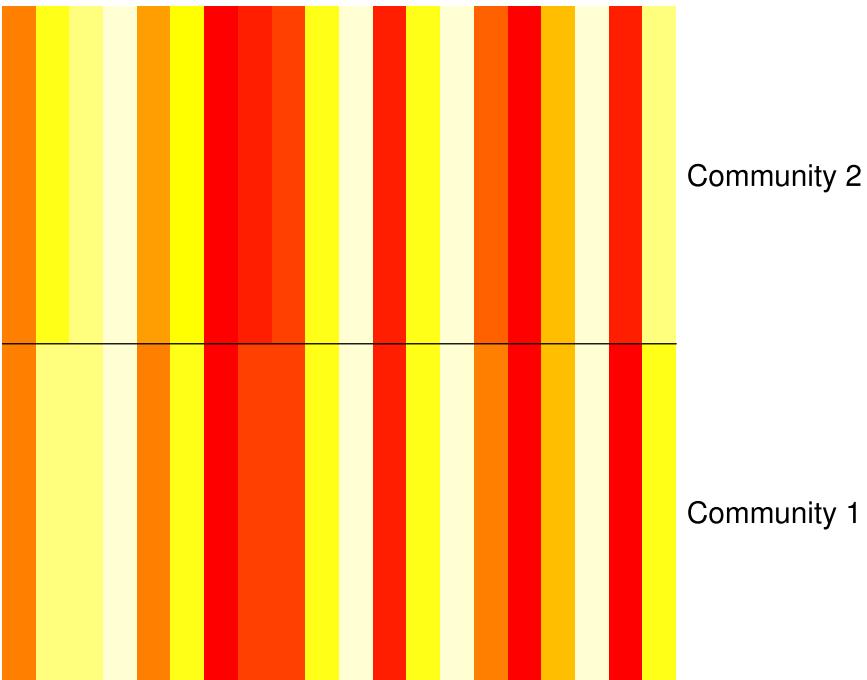


Fig. 12. Visualization of the learned structure representations of two similar communities.

6 RELATED WORK

Representation learning. Our work has connections with representation learning that can be categorized into three main approaches: (i) probabilistic models, (ii) geometrically motivated manifold-learning approaches, and (iii) reconstruction-based algorithms related to auto-encoder.

The key idea of the probabilistic model-based approaches is to use unsupervised feature learning to learn a hierarchy of features one level at a time [6, 28, 35, 45, 48]. For example, Wang et al. used a regression learner to learn the optimized layout of heterogeneous elements on the search result page (SERP) [51]. The work in Reference [2] used an unsupervised learning method to obtain a hierarchy of features one level at a time and to learn a new transformation at each level to be composed with the previously learned transformations.

In the second category, the large majority of the algorithms adopt a non-parametric approach, based on a training set nearest-neighbor graph [7, 43, 47, 52, 52]. Hinton et al. [27] and Bengio et al. [4] exploited the RBMs to perform unsupervised feature learning for natural image modeling. The work in Reference [43] introduced “t-SNE” that was built on a geometric perspective that adopts a non-parametric approach, based on a training set nearest-neighbor graph, which is a variation of Stochastic Neighbor Embedding [26].

As for the auto-encoder-based methods, compared to probabilistic models, it does not need complicated posterior distributions because of the use of latent variables. Auto-encoders can directly parameterize features or representation functions and learn a direct encoding [3, 5, 30, 34, 56, 63]. Therefore, we chose the auto-encoder method as our base model and further developed a collective spatiotemporal auto-encoder to learn the representation of community structure.

Urban computing. Urban computing is a process aiming to tackle major issues in cities by analyzing and modeling urban data (e.g., traffic flow, human mobility, and geographical data). One of

the biggest challenges in urban computing is to compute with heterogeneous data [61]. Zheng et al. proved that setting equally weight for different data source in a regression or classification model does not achieve the best performance [62]. Yuan et al. discovered regional functions of a city using POIs and taxi traces [59]. Zhang et al. first detected spatiotemporal hotspots and then from geo-tagged social media data and then used both reconstruction- and single graph-based strategies to learn the representations of geo-tagged time-stamped words [60]. Zhang et al. employed an accelerated mode seeking procedure to detect spatial-temporal hot spots underlying people's activities and jointly embedded all spatial, temporal, and textual units into the same space [60]. Compared to the embeddings of geo-tagged time-stamped words, our work targets at a different spatial granularity and aims at learning the representation of an entire urban community. Fu et al. proposed a probabilistic latent factor model to learn the portfolios of urban functions in a zone [18]. Cici et al. identified emerging patterns with multi-relational approach from spatial data [11]. Wang et al. adopted the skip-gram model to learn the region representation from urban and mobile data [50]. Different from the skip-gram model on single graph, our work focuses on collectively learning from multiple spatiotemporal graphs. Bejan et al. mined the driving route for end users by considering physical features such as route, traffic flow, and driving behavior [1]. Liu et al. detected spatial-temporal causality of outliers in traffic data [40]. Liu et al. provided an integrated mobility pattern analysis between the location traces of taxicabs and the mobility records in bus transactions [41]. Lan et al. introduced a road segment-based anomaly detection problem that detects abnormal road segments, each of which has its "real" traffic deviating from its "expected" traffic and infers the major causes of anomalies on the road network [33]. Liu et al. focused on the identification and optimization of flawed region pairs with problematic bus routing to improve utilization efficiency of public transportation services, according to people's real demand for public transportation [42]. Liu et al. provided a focused study of temporal retweeting patterns and their influence on social media marketing campaigns [38]. Yao et al. presented a novel method that incorporates the degree of temporal matching between users and POIs into personalized POI recommendations [58]. Fu et al. developed a system, named CUMMA, for classifying service usages of mobile messaging apps by jointly modeling user behavioral patterns, network traffic characteristics, and temporal dependencies [20]. Liu et al. proposed a bike-sharing network optimization approach by considering multiple influential factors to enhance the quality and efficiency of the bike-sharing service by selecting the right station locations [39]. Yao et al. proposed a Deep Multi-View Spatial-Temporal Network (DMVST-Net) framework to model both spatial and temporal relations [57].

Learning to rank. In addition, our work is related to the Learning-to-Rank method, which includes pointwise, pairwise, and listwise approaches. The pointwise methods [24] reduce the LTR task to a regression problem: Given a single query-document pair, predict its score. The pairwise methods approximate the LTR task to a classification problem. The goal of the pairwise ranking is to learn a binary classifier to identify the better document in a given document pair by minimizing the average number of inversions in ranking [8, 16]. The listwise methods optimize a ranking loss metric over lists instead of document pairs [53]. For instance, Li et al. proposed AdaRank [55] and ListNet [10], and Burges et al. proposed LambdaMART [9]. More recent work in Reference [32] further learned the ranking model that is constrained to be with only a few nonzero coefficients using L1 constraint and propose a learning algorithm from the primal dual perspective.

7 CONCLUSION REMARKS

In this article, we studied the problem of learning urban community structures. We take into account not only the points of interests but also human movement among these POIs. We formulate the problem as a learning task over multiple mobility graphs of POIs and propose a novel

collective embedding framework. The framework consists of three major steps. We started with a probabilistic propagation method to unify and represent static POIs and dynamic human mobility records as periodic spatial-temporal mobility graphs. We then developed a collective embedding method to learn the embeddings of POIs from the obtained mobility graphs. Based on the POIs embeddings, we further proposed an unsupervised graph-based weighted aggregation method to identify community embeddings. To evaluate the performance of the proposed approach, we applied it to predict WTP for communities and spot vibrant communities from real datasets. The experimental results show that our approach can effectively learn the representation of community structures and substantially enhance the vibrant community prediction accuracy. Finally, it is worth noting that our proposed collective framework also has the potential to be generalized to learn the structural representations of other geographic items.

REFERENCES

- [1] Andrei Iu Bejan, Richard J. Gibbens, David Evans, Alastair R. Beresford, Jean Bacon, and Adrian Friday. 2010. Statistical modelling and analysis of sparse bus probe data in urban areas. In *Proceedings of the 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC'10)*. IEEE, 1256–1263.
- [2] Yoshua Bengio and others. 2009. Learning deep architectures for AI. *Found. Trends Mach. Learn.* 2, 1 (2009), 1–127.
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1798–1828.
- [4] Yoshua Bengio, Aaron C. Courville, and James S. Bergstra. 2011. Unsupervised models of images by spike-and-slab RBMs. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. 1145–1152.
- [5] Yoshua Bengio and Olivier Delalleau. 2009. Justifying and generalizing contrastive divergence. *Neur. Comput.* 21, 6 (2009), 1601–1621.
- [6] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS'07)*. 153–160.
- [7] Matthew Brand. 2003. Charting a manifold. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems (NIPS'03)*. 985–992.
- [8] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 89–96.
- [9] Christopher J. C. Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23–581 (2010), 81.
- [10] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML'07)*.
- [11] Michelangelo Ceci, Annalisa Appice, and Donato Malerba. 2007. Discovering emerging patterns in spatial databases: A multi-relational approach. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 390–397.
- [12] Ye Ding, Yanhua Li, Ke Deng, Haoyu Tan, Mingxuan Yuan, and Lionel M. Ni. 2017. Detecting and analyzing urban regions with high impact of weather change on transport. *IEEE Trans. Big Data* 3, 2 (2017), 126–139.
- [13] William E. Donath and Alan J. Hoffman. 1973. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.* 17, 5 (1973), 420–425.
- [14] Casey Dougal, Christopher A. Parsons, and Sheridan Titman. 2015. Urban vibrancy and corporate growth. *J. Financ.* 70, 1 (2015), 163–210.
- [15] Jordi Duch and Alex Arenas. 2005. Community detection in complex networks using extremal optimization. *Phys. Rev. E* 72, 2 (2005), 027104.
- [16] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.* 4, (Nov. 2003), 933–969.
- [17] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* (2001), 1189–1232.
- [18] Yanjie Fu, Guannan Liu, Spiros Papadimitriou, Hui Xiong, Yong Ge, Hengshu Zhu, and Chen Zhu. 2015. Real estate ranking via mixed land-use latent models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 299–308.
- [19] Yanjie Fu, Hui Xiong, Yong Ge, Zijun Yao, Yu Zheng, and Zhi-Hua Zhou. 2014. Exploiting geographic dependencies for real estate appraisal: A mutual perspective of ranking and clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1047–1056.

- [20] Yanjie Fu, Hui Xiong, Xinjiang Lu, Jin Yang, and Can Chen. 2016. Service usage classification with encrypted internet traffic in mobile messaging apps. *IEEE Trans. Mobile Comput.* 15, 11 (2016), 2851–2864.
- [21] Edward L. Glaeser, Jed Kolko, and Albert Saiz. 2001. Consumer city. *J. Econ. Geogr.* 1, 1 (2001), 27–50.
- [22] Roger Guimera and Luis A. Nunes Amaral. 2005. Functional cartography of complex metabolic networks. *Nature* 433, 7028 (2005), 895.
- [23] Roger Guimera, Marta Sales-Pardo, and Luís A. Nunes Amaral. 2004. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* 70, 2 (2004), 025101.
- [24] LI Hang. 2011. A short introduction to learning to rank. *IEICE Trans. Inf. Syst.* 94, 10 (2011), 1854–1862.
- [25] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. 2009. Boosting and additive trees. The elements of statistical learning. (2009).
- [26] Geoffrey Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, Vol. 15. 833–840.
- [27] Geoffrey E. Hinton and others. 2010. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2551–2558.
- [28] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neur. Comput.* 18, 7 (2006), 1527–1554.
- [29] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2009. Replicated softmax: An undirected topic model. In *Advances in Neural Information Processing Systems*. 1607–1614.
- [30] Geoffrey E. Hinton and Richard S. Zemel. 1994. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in Neural Information Processing Systems*. 3–10.
- [31] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 7 (2002), 881–892.
- [32] Hanjiang Lai, Yan Pan, Cong Liu, Liang Lin, and Jie Wu. 2013. Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Trans. Comput.* 62, 6 (2013), 1221–1233.
- [33] Jinsong Lan, Cheng Long, Raymond Chi-Wing Wong, Youyang Chen, Yanjie Fu, Danhuai Guo, Shuguang Liu, Yong Ge, Yuanchun Zhou, and Jianhui Li. 2014. A new framework for traffic anomaly detection. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 875–883.
- [34] Hugo Larochelle and Yoshua Bengio. 2008. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 536–543.
- [35] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 609–616.
- [36] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1099–1108.
- [37] Zhenhui Li, Jingjing Wang, and Jiawei Han. 2012. Mining event periodicity from incomplete observations. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 444–452.
- [38] Guannan Liu, Yanjie Fu, Tong Xu, Hui Xiong, and Guoqing Chen. 2014. Discovering temporal retweeting patterns for social media marketing campaigns. In *Proceedings of the 2014 IEEE International Conference on Data Mining (ICDM'14)*. IEEE, 905–910.
- [39] Junming Liu, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Hui Xiong, Hao Zhong, and Yanjie Fu. 2015. Station site optimization in bike sharing systems. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM'15)*. IEEE, 883–888.
- [40] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. 2011. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1010–1018.
- [41] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu, and Junjie Wu. 2014. Exploiting heterogeneous human mobility patterns for intelligent bus routing. In *2014 IEEE International Conference on Data Mining (ICDM'14)*. IEEE, 360–369.
- [42] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu, and Junjie Wu. 2017. Intelligent bus routing with heterogeneous human mobility patterns. *Knowl. Inf. Syst.* 50, 2 (2017), 383–415.
- [43] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, (Nov. 2008), 2579–2605.
- [44] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

- [45] Christopher Poulney, Sumit Chopra, Yann L. Cun, and others. 2007. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*. 1137–1144.
- [46] Matthew J. Rattigan, Marc Maier, and David Jensen. 2007. Graph clustering with network structure indices. In *Proceedings of the 24th International Conference on Machine Learning*. ACM, 783–790.
- [47] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.
- [48] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep boltzmann machines. In *Artificial Intelligence and Statistics*. 448–455.
- [49] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W. Schuller. 2017. A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 3 (2017), 417–429.
- [50] Hongjian Wang and Zhenhui Li. 2017. Region Representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM'17)*. ACM, New York, NY, 237–246.
- [51] Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. ACM, 103–112.
- [52] Kilian Q. Weinberger and Lawrence K. Saul. 2006. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* 70, 1 (2006), 77–90.
- [53] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 1192–1199.
- [54] Ren Xingyi, Song Meina, E Haihong, and Song Junde. 2016. Textual-geographical-social aware point-of-interest recommendation. *J. Chin. Univ. Posts Telecommun.* 23, 6 (2016), 24–67.
- [55] Jun Xu and Hang Li. 2007. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*.
- [56] LE Yann. 1987. *Modèles connexionnistes de l'apprentissage*. Ph.D. Dissertation. These de Doctorat, Universite Paris 6.
- [57] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. *arXiv preprint arXiv:1802.08714* (2018).
- [58] Zijun Yao, Yanjie Fu, Bin Liu, Yanchi Liu, and Hui Xiong. 2016. POI recommendation: A temporal matching between poi popularity and user regularity. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 549–558.
- [59] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 186–194.
- [60] Chao Zhang, Keyang Zhang, Quan Yuan, Haoruo Peng, Yu Zheng, Tim Hanratty, Shaowen Wang, and Jiawei Han. 2017. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 361–370.
- [61] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 38.
- [62] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1436–1444.
- [63] Will Y. Zou, Andrew Y. Ng, and Kai Yu. 2011. Unsupervised learning of visual invariance with temporal coherence. In *Proceedings of the Neural Information Processing Systems 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, Vol. 3.

Received December 2017; revised March 2018; accepted April 2018