Ensemble-Spotting: Ranking Urban Vibrancy via POI Embedding with Multi-view Spatial Graphs

Pengyang Wang [*] Jiawei Zhang [†] Guannan Liu[‡] Yanjie Fu[+§] Charu Aggarwal[¶]

Abstract

Vibrant residential communities are defined as places with permeability, vitality, variety, accessibility, identity and legibility. Developing vibrant communities can help boost commercial activities, enhance public security, foster social interaction, and thus yield livable, sustainable, and viable environments. However, it is challenging to understand the underlying drivers of vibrant communities to make them traceable and predictable. Toward this goal, we study the problem of ranking vibrant communities using human mobility data and point-of-interests (POIs) data. We analyze large-scale urban and mobile data related to residential communities and find that in order to effectively identify vibrant communities, we should not just consider community "contents" such as buildings, facilities, and transportation, but also take into account the *spatial structure*. The spatial structure of a community refers to how the geographical items (POIs, road networks, public transits, etc.) of a community are spatially arranged and interact with one another. Along this line, we first develop a geographical learning method to find proper representations of communities. In addition, we propose a novel geographic ensemble ranking strategy, which aggregates a variety of weak rankers to effectively spot vibrant communities. Finally, we conduct a comprehensive evaluation with real-world residential community data. The experimental results demonstrate the effectiveness of the proposed method.

Keywords: POI Embedding, Multi-view Graph, Ensemble Ranking

1 Introduction

Vibrant communities are defined as residential places with permeability, vitality, variety, accessibility, identity and legibility. Designing vibrant communities can help boost commercial activities, enhance public security, foster social interaction, and thus yield livable, sustainable, and viable environments. Modern sensing and Web technologies have enabled the collection of a variety of big crowd-sourced geo-tagged data (BCGD). For example, urban geography data includes point-of-interests such as restaurants, banks, and schools. Meanwhile human mobility data can be accessed through the GPS trajectories of the vehicles, the mobile devices, the check-in of their footprints, and the geo-tagged posts in social media. The abundant BCGD provides invaluable insights for understanding human mobility patterns.

In this paper, we develop a geographic learning method from BCGD to mimic and spot the geospatial patterns of vibrant communities. We analyze a large number of residential communities and found that the vibrancy of communities follows a power-law distribution. In other words, only a small number of communities are highly vibrant, while most communities are not. Therefore, we propose a method to recognize the patterns of highly-vibrant communities by formulating the problem as that of ranking the vibrancy of communities by leveraging human mobility and point-of-interests data.

A vibrant community is not only abundant in the number of POIs, but also boasts appropriate POI layouts and *spatial structure* [22]. The spatial structure defines how the geographical items (POIs, road networks, public transits, etc.) of a community are spatially arranged and interact with one another. Figure 1 shows two different communities with differing levels of vibrancy. The density of POIs in the community with higher vibrancy (cf. Figure 1(a)) is much larger than that in Figure 1(b). Moreover, almost all POIs in Figure 1(a) are located near roads and highly interconnected. Observe that the spatial structures of the communities vary greatly, which is often a unique indicator for community vibrancy.

However, the spatial structure of communities has not been sufficiently explored in prior works for paucity of data containing quantitative measurements of the geographical layout. In this study, we tackle this problem from a perspective of representation learning. In particular, for each community, we first exploit

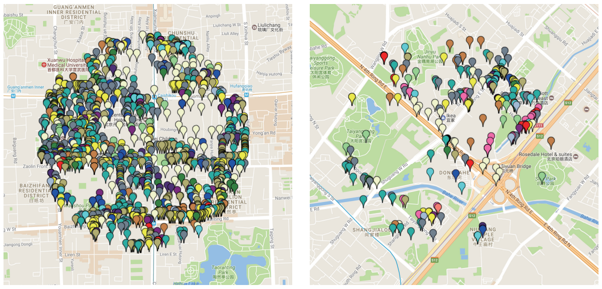
[*+] Missouri S&T, MO, USA. Email: {pwqt3, fuyan}@mst.edu.

[†] Florida State University, FL, USA. Email: jzhang@cs.fsu.edu.

[‡] Beihang University, BJ, China. Email: liugn@buaa.edu.cn.

[§] Contact Author.

[¶] IBM T. J. Watson Research Center, NY, USA. Email: charu@us.ibm.com.



(a) A High Vibrant Community (b) A Low Vibrant Community

Figure 1: Structure Comparison between High and Low Vibrant Communities

the POIs of the community to construct two spatial graphs from two views of (i) geographic distance and (ii) mobility connectivity. The nodes of both graphs stand for POIs in the community. In the *distance graph*, the weight of each edge denotes the geographic distance between two corresponding POIs. In the *connectivity graph*, the weight of each edge denotes the mobility connectivity between two corresponding POIs. We develop a mobility propagation based method to infer the connectivity strength between a POI pair. In this way, the spatial structure of the community can be represented by these two graphs.

In order to represent the structures of the spatial graphs, we propose dynamic geographical embedding algorithms to learn latent vector representations of communities. Specifically, we propose to aggregate and convert both distance and connectivity graphs at the POI level to new distance and connectivity graphs at a higher level of POI categorization. Since the nodes of these new spatial graphs represent POI categories, the sizes of all community spatial graphs will be the same. Finally, we develop an autoencoder-based graph embedding method to learn the vector representations of spatial structures of each community. Moreover, we combine the latent features of spatial structure with the explicit features of buildings and infrastructure.

In addition to leveraging spatial structure for representation learning, we propose an accurate geographical ranking model to learn, mimic, and spot the patterns of highly-vibrant communities. Although traditional document ranking methods can be adapted, their performance is usually limited because of the inability to account for spatial autocorrelations and geographical dependencies. To improve performance, we propose a geographical ensemble ranking method based on non-negative matrix factorization. The proposed geographic ensemble ranking method is capable of combining the ranking scores of multiple weak point-wise, pair-wise, and list-wise rankers, and produces a more accurate and robust ranking list for spotting vibrant communities.

In summary, we first develop a geographical learning method for finding community representations and combine it with an ensemble ranking model for spotting vibrant communities. Specifically, we first construct spatial graphs for each community from multiple views and encode these spatial graphs into latent vectors of communities. In addition, we propose a novel geographic ensemble ranking strategy to aggregate a variety of weak rankers for effectively spotting vibrant communities with the learned community representations as model inputs. Finally, we conduct a comprehensive evaluation with real-world residential community data. The experimental results demonstrate the effectiveness of the proposed method.

2 Preliminaries

We first introduce some important definitions, and then present an overview of the proposed method.

2.1 Definitions

Definition 1. Residential Community. In this study, a residential community consists of a location (i.e., latitude and longitude) of a residential complex and a neighborhood area (e.g., a circle with radius of $1km$). There could be various POIs in the neighborhood area, providing many services to people.

2.2 Framework Overview

Figure 3 shows an overview of our proposed method. There are two specific tasks: (i) learning discriminative representations of communities, and (ii) developing an accurate geographic ranking indicator to mimic the patterns of highly-vibrant communities. In the first task, we learn the representations of urban communities. In particular, we first extract explicit features of POIs distribution and human mobility for each community. To extract the latent structural features of communities, we present a spatial graph embedding method that includes spatial graph construction and a collective spatial graph autoencoder. In the second task, we first create a variety of pointwise, pairwise, and listwise rankers, and then design a non-negative matrix factorization based ensembling method to aggregate the results of weak rankers into final rankings of community vibrancy.

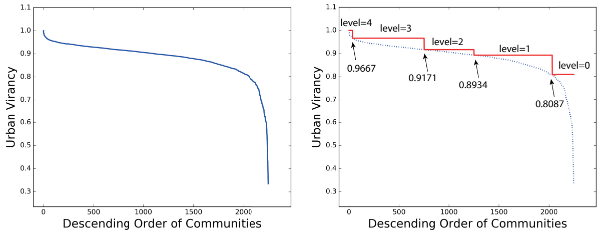
3 Quantifying Urban Vibrancy

In prior literatures, researchers have developed a conceptual and empirical understanding that community vibrancy can be reflected by consumer activities from two perspectives: density and diversity of consumer activities [19].

Along this line, we propose *a fused scoring framework* to quantify urban vibrancy. First, we quantify the density using the total number of mobile check-in events as an estimation, denoted by $freq$. Then, we quantify the diversity of consumer activities by exploiting the entropy measurement over the number of mobile check-in events with respect to different POI categories: $div = \sum_i^{\mathbb{C}} x_i \log x_i$, where x_i represents the number of check-in events of the i^{th} POI category and $\mathbb{C}$ denotes the number of POI categories. Finally, we use the harmonic mean, $vibrancy = \frac{2 \times freq \times div}{freq + div}$ to fuse both density and diversity into a single score.

In this paper, we calculate the "community vibrancy" based on the proposed fused scoring framework for each community for Beijing city. Then, all the communities are sorted in a descending order in terms of the computed vibrancy. Figure 2(a) shows the fact consistent with our common sense that few points in the city representing centers of the city may attract many people to visit and consume, while most communities play a mediocre role in our daily life.

Besides, we split the curve of Figure 2(b) into five segments using four inflection points representing vibrancy values 0.9667, 0.9171, 0.8934 and 0.8087 respectively. Hence, we can assign five-level ratings to each segment as its ranking relevance label, ranging from 0 to 4, shown in Figure 2(b).



(a) Analysis of Vibrant Communities (b) Relevance Level of Vibrant Communities

Figure 2: Analysis of Urban Residential Community Vibrancy

4 Representation Learning of Residential Communities

We present a spatial graph embedding based framework for representation learning of residential communities.

4.1 Explicit Features for Community Representation We extract explicit features from POI data and taxi trace records. Based on the data sources we used, the explicit features of communities are extracted by two categories: *spatial related features* from POI statics data and *human mobility related features* from taxi trace records. For *spatial related features*, we use two kinds of statistics: (1) POI numbers per category, and (2) average distance between POIs. For *human mobility related features*, we use three kinds of statistics: (1) average speed of mobility traces, (2) average mobility distance, and (3) the amount of mobility traces. The details about explicit features are shown in Table 2.

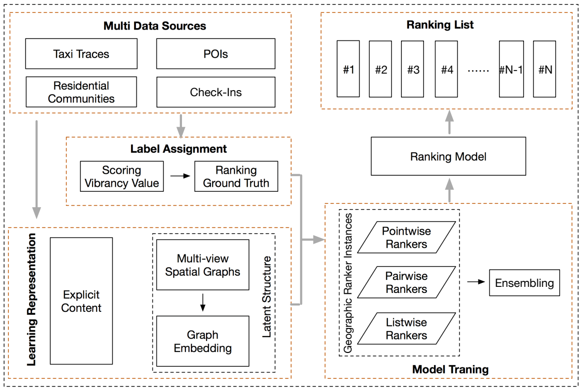


Figure 3: Framework Overview

4.2 Constructing Multi-view Spatial Graphs of Communities We analyze large-scale urban and mobile data and find that community vibrancy is not only impacted by community "contents" such as buildings, facilities, transportations, but also influenced by community "layouts" : *spatial structure*. As a result, aside from extracting explicit features of communities, we need to take into account spatial structure for improving the representation learning of communities.

We tackle this problem from the graph perspective, where we first represent a community with spatial graphs and then learn the representation of spatial structures via graph embedding. Intuitively, for each community, we can regard POIs as nodes in a graph and then construct: (1) geographic distance graph, in which the weight of each edge represents geographic distance between two POIs and (2) mobility connectivity graph, in which the weight of each edge represents the mobility connectivity between two POIs. Unfortunately, different communities might have different numbers of POIs, and therefore the sizes of graphs vary over communities. Many existing graph embedding techniques cannot be easily converted to conduct dynamic graph embedding for varying graph sizes. To control and fix the graph size, we further propose to aggregate and convert the two POI-level graphs to two new graphs at POI category level. Since the number of POI categories is fixed, and the nodes of the new graphs represent POI categories, such graphs are of a fixed size. Next, we detail how to construct the distance and mobility graphs at POI category level.

Geographical distance graph across POI categories. The relative geographic distances among POIs in a community can reflect the spatial structure and configuration of this community via the spatial allocation of buildings. To construct the geographical distance graph for a community, we calculate the average distances between POI category pairs. Specifically, we first pick the most important POI categories as shown in Table 1. We then categorize all the POIs into 20 groups in terms of

Table 1: POI Categories

Number	POI Category Name	Number	POI Category Name
1	Vehicle Service	11	Tourist
2	Car Dealer	12	Real Estate
3	Repair & Maintenance	13	Government & Non-Government Organization
4	Motorbike Dealer & Service	14	Culture & Education
5	Food & Beverage	15	Transportation
6	shopping	16	Finance & Insurance
7	Daily Life Service	17	Company & factory
8	Sports & Recreation	18	Road Furniture
9	Medical Service	19	Named Place & Address
10	Lodging	20	Public Service

the 20 POI categories.

Later, the average distance between the m^{th} POI category and the n^{th} POI category can be calculated by

$$\overline{dist}_{m,n} = \frac{\sum_{j\in\Theta_m}\sum_{i\in\Theta_n} dist_{i,j}}{|\Theta_m||\Theta_n|} \tag{4.1}$$

where i and j are two distinct POIs. Θ_m and Θ_n are two POI sets of the m^{th} category and the n^{th} category, respectively. $dist_{i,j}$ is the distance between i and j. Finally, we denote the geographic distance graph with $\mathcal{D} \in \mathbb{R}^{N\times N}$, where N is the number of POI categories and each element of $\mathcal{D}$ is the average distance between two corresponding POI categories.

Mobility connectivity graph across POI categories. People's outdoor activities include the transitions from one POI to another POI, and ultimately form mobility flows in a community. As a result, human mobility can indicate the connectivity among POIs. We propose a four-step algorithm to learn the mobility connectivity between two POI categories from taxicab GPS trajectory data. In particular, we estimate the possibility of passengers moving from one POI category to another POI category by exploiting a propagation based method in [13].

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 road network distance between the drop-off point d and the POI p:

$$P(x) = \frac{\beta_1}{\beta_2} \cdot x \cdot exp(1 - \frac{x}{\beta_2}), \tag{4.2}$$

where $\beta_1 = \max_x P(x)$ and $\beta_2 = \arg\max_x P(x)$. Specifically, when $x = 0$, $P(x) = 0$. Since a taxi may not send passengers into a POI directly, the drop-off point usually is not the same as the destination, but usually close to the destination. A passenger often walks a short distance to reach the destination. Hence, when the distance exceeds a threshold β_2, the probability keeps decreasing with an exponential heavy tail. With this function, we can propagate the visit probability of a passenger from the drop-off point to its surrounding POIs.

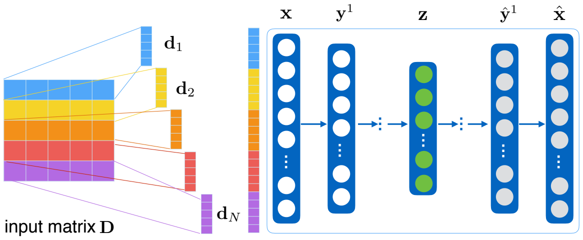


Figure 4: The Framework for Latent Feature Extraction

Step 2: Calculate POI-level visit probability. We aggregate all probabilities from all drop-off points in taxi traces: $\tau(p) = \sum_{d\in\mathcal{D}} P(dis(d,p))$, where $\mathcal{D}$ is the drop-off point set of taxi traces in the community.

Step 3: Calculate POI-category-level visit probability. For each POI category i in a community, the POI category-level aggregated visit probability is given by: $\phi_i = \sum_{p\in i} \tau(p)$, where $p \in i$ denotes the POI p belongs to the i^{th} POI category.

Step 4: Calculate visit flow probabilities between categories. We calculate the flow probabilities from the i^{th} POI category to the j^{th} POI category:

$$\phi_{\vec{ij}} = \begin{cases} \phi_i \cdot \phi_j, & if \ \ i \neq j \\ 0, & if \ \ i = j \end{cases} \tag{4.3}$$

Finally, we can obtain a mobility connectivity graph across POI categories, denoted by a matrix $\mathcal{P} \in \mathbb{R}^{N\times N}$ where N is the number of POI categories, and $\mathcal{P}[i,j] = \phi_{\vec{ij}}$ denotes the mobility connectivity from the i^{th} POI category to the j^{th} POI category.

4.3 Learning Spatial Structure via Spatial Graph Embedding

The *explicit features* extracted in the previous subsection effectively illustrate the basic statistical information about the communities in terms of "contents". However, many other non-explicit information about the communities can hardly be captured with these *explicit features*, like *spatial structure*. Formally, the *spatial structure* denotes how the geographic items (POIs, road networks, public transits, etc.) of a community are spatially displayed, configured, and interacted, such as locations, types, interconnectivity, or any other structural properties. In this part, we will represent them with a latent feature vector extracted from the *geographic distance graph* and *mobility connectivity graph* involving the POI categories in the community.

4.3.1 Deep Auto-encoder Model

The *latent feature* extraction process is based on the deep auto-encoder model[2] . 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. Figure 4 shows that auto-encoder model involves two steps: encode and decode. The encode part projects the original feature vector to the objective feature space, while the decode step recovers the latent feature representation to a reconstruction space. In auto-encoder model, we generally need to ensure that the original feature representation of instances should be as similar to the reconstructed feature representation as possible.

Formally, let $\mathbf{x}$ represent the original feature representation of instance i, and $\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^o$ be the latent feature representations of the instance at hidden layers $1, 2, \cdots, o$ in the encode step respectively, the encoding result in the objective lower-dimension feature space can be represented as $\mathbf{z} \in \mathbb{R}^d$ with dimension d. Formally, the relationship between these vector variables can be represented with the following equations:

$$(4.4)\quad \begin{cases} \mathbf{y}^1 & = \sigma(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1), \\ \mathbf{y}^k & = \sigma(\mathbf{W}^k\mathbf{y}^{k-1} + \mathbf{b}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \mathbf{z} & = \sigma(\mathbf{W}^{o+1}\mathbf{y}^o + \mathbf{b}^{o+1}). \end{cases}$$

Meanwhile, in the decode step, the input will be the latent feature vector $\mathbf{z}$ (i.e., the output of the encode step), and the final output will be the reconstructed vector $\hat{\mathbf{x}}$. The latent feature vectors at each hidden layers can be represented as $\hat{\mathbf{y}}^o, \hat{\mathbf{y}}^{o-1}, \cdots, \hat{\mathbf{y}}^1$. The relationship between these vector variables can be denoted as

$$(4.5)\quad \begin{cases} \hat{\mathbf{y}}^o & = \sigma(\hat{\mathbf{W}}^{o+1}\mathbf{z} + \hat{\mathbf{b}}^{o+1}), \\ \hat{\mathbf{y}}^{k-1} & = \sigma(\hat{\mathbf{W}}^k\hat{\mathbf{y}}^k + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \hat{\mathbf{x}} & = \sigma(\hat{\mathbf{W}}^1\hat{\mathbf{y}}^1 + \hat{\mathbf{b}}^1). \end{cases}$$

In the above equations, $\mathbf{W}$s and $\mathbf{b}$s denote the weight matrices and bias terms to be learned in the model. The objective of the auto-encoder model is to minimize the loss between the original feature vector $\mathbf{x}$ and the reconstructed feature vector $\hat{\mathbf{x}}$. Formally, the loss term can be represented as

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 .$$

4.3.2 Latent Feature Extraction

Spatial structure denotes the distribution of POIs inside the community, e.g., a grocery store lies between two residential buildings; a school is next to the police office. The *Spatial structure* can hardly be represented with explicit features extracted before, and we propose to represent them with a set of latent feature vectors extracted from the *geographic distance graph* and the *mobility connectivity graph* defined in the previous subsection. The auto-encoder model is applied here for the latent feature extraction.

Auto-encoder model has been applied to embed the graph data into lower-dimensional spaces in many of the research works, which will obtain a latent feature representation for the nodes inside the graph. Different from these works, instead of calculating the latent feature for the POI categories inside the communities, we aim at obtaining the latent feature vector for the whole community, i.e., embedding the graph as one latent feature vector.

As shown in Figure 4, we transform the matrix of the *geographical distance graph* (involving the POI categories) $\mathcal{D}$ into a series of vectors $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N \in \mathbb{R}^{N \times 1}$, where vector $\mathbf{d}_i$ denotes the i^{th} row of matrix $\mathcal{D}$, i.e., $\mathcal{D}(i, :)$. These vectors are appended via concatenation and the resulting vector can be represented as $\mathbf{d} = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \cdots, \mathbf{d}_N^\top]^\top \in \mathbb{R}^{N^2 \times 1}$, which will be used as the input feeding into the auto-encoder model. The latent embedding feature vector of $\mathbf{d}$ can be represented as $\mathbf{x}_D$ (i.e., the vector $\mathbf{z}$ as introduced in the auto-encoder model in the previous section), which depicts the layout information of POI categories in the community in terms of the geographical distance.

Besides the static layout based on *geographic distance graph*, the spatial structure of the POIs in the communities can also be revealed indirectly through the human mobility. For a pair of POI categories which are far away geographically, if people like to go between them frequently, it can display another type of structure of the POIs in terms of their functional correlations. Therefore, the mobility connectivity graph involving the POIs, i.e., matrix $\mathcal{P}$, is also applied for latent feature extraction, and the resulting embedding feature vector can be represented as $\mathbf{x}_P$.

For a given community, in addition to the *explicit feature vector* $\mathbf{x}_E$ extracted from the previous section, these two embedding feature vectors $\mathbf{x}_D$ and $\mathbf{x}_P$ are used as the *latent feature vectors* illustrating the community layout information. Formally, we can represent all the extracted features for a community as $\mathbf{x} = [\mathbf{x}_E^\top, \mathbf{x}_D^\top, \mathbf{x}_P^\top]^\top$, which will be used to train the ranking model to determine the communities vibrancy.

In a nutshell, we summarize all the explicit and latent learned features in Table 2

5 Ensemble Geographic Ranking for Spotting Vibrant Communities

To predict ranking of communities on vibrancy value, We propose a method to ensemble multiple weak rankers to achieve a more accurate and stable ranker.

5.1 Creating Geographic Ranker Instances

We use Learning to rank (LTR) algorithms to generate geographic ranker instances. LTR algorithms are firstly developed for ranking items generated by the search engine based on queries. LTR algorithms treat each item as a document and rank these documents based on the relevance between documents and queries. For our task, we regard each community as a document represented by explicit features and latent features. There are three kinds of learning to rank (LTR) algorithms in the literature, including pointwise rankers, like Random Forest [16]; pairwise rankers, like Rankboost [8]; listwise rankers, like AdaRank [24]. We generate multiple LTR ranker instances by setting different attributes for each LTR algorithms.

Table 2: Feature Summary

Feature Type	Input Data Views	Feature
Explicit	Spatial distribution of POIs	POI number per category average distance between POIs
	Human mobility traces	average speed of mobility average mobility distance the amount of movements
Latent	Spatial distribution of POIs	Geographical distance graph embedding feature vector
	Human mobility traces	Mobility connectivity graph embedding feature vector

5.2 Ensembling Geographic Ranking Each geographic ranker instance provides a ranking score for each community and the results from different rankers could be very divergent. We now present a factorization technique based on non-negative matrix factorization to ensemble the ranking results into a consent one. In this case, all rankers can achieve a consensus on how they assess a community.

Let $\mathbf{\Delta}_{r,\chi}$ be the ranking score outputted by the ranker r for the community χ, the ranking score is normalized across communities to be in the range $[0, 1]$. Take the matrix of ranking score $\mathbf{\Delta} \in [0, 1]^{t\times g}$, where t is the number of geographic ranker instances and g is the number of communities. Then, we perform a non-negative matrix factorization on $\mathbf{\Delta}$: $\mathbf{\Delta} \approx \mathbf{\Lambda}\mathbf{\Omega}^T$, where $\mathbf{\Lambda} \in \mathbb{R}^{t\times k}$ and $\mathbf{\Omega} \in \mathbb{R}^{g\times k}$, by minimizing the Kullback-Leibler divergence between $\mathbf{\Delta}$ and $\mathbf{\Lambda}\mathbf{\Omega}^T$ [7]:

$$(5.6)\quad \min_{\mathbf{\Lambda},\mathbf{\Omega}\geqslant 0} \sum_{r,\chi} \mathbf{\Delta}_{r,\chi} \log \frac{\mathbf{\Delta}_{r,\chi}}{\mathbf{\Lambda}_{r,*}\mathbf{\Omega}_{\chi,*}} - \mathbf{\Delta}_{r,\chi} + \mathbf{\Lambda}_{r,*}\mathbf{\Omega}_{\chi,*}$$

Equation 5.6 shows a general view of factorization. When we set $k = 1$, the optimization results in ensembling all rankers into a single perspective.

6 Experimental Results

We provide an empirical evaluation of the performances of the proposed method on real-world data.

6.1 Experimental Data Table 3 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. Also, we extract POIs related data from www.dianping.com which is a business review site in China. Moreover, we crawl the Beijing residential community data from www.soufun.com which is the largest real-estate online system in China. Furthermore, the check-in data of Beijing is crawled from www.jiepang.com which is a Chinese version of Fourquare. Each check-in event includes name, category, address, longitude and latitude of POIs.

Table 3: 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

6.2 Baseline Algorithms To show the effectiveness of our method, we compare our method against the following combinations of features and LTR algorithms. First, we give a brief introduction of the baseline algorithms. (1) *MART* [10]: it is a boosted tree model, specifically, a linear combination of the outputs of a set of regression trees. (2) *RankBoost* [8]: it is a boosted pairwise ranking method, which trains multiple weak rankers and combines their outputs as final ranking. (3) *AdaRank* [24]: it is a listwise learning algorithm within the framework of boosting, which can minimize a listwise loss function. (4) *Random Forests* [16] : it is a ranking strategy through learning the predictions from an ensemble of random trees. Then, we respectively combine explicit feature, latent features or explicit+latent features with the above single rankers as follows. (1)Latent Features combined with RankBoost (**LF&RankBoost**). (2)Explicit+Latent Features combined with MART (**ELF&MART**). (3)Explicit+Latent Features combined with *AdaRank* (**ELF&AdaRank**). (4)Explicit+Latent Features combined with *RandomForest* (**ELF&RandomForests**). Besides, we also combine the proposed ensemble geographic ranking method with latent features (**LF&ER**) or explicit features (**EF&ER**), respectively.

We utilize RTree to index geographic items (i.e., taxi and bus trajectories, check-ins, etc.) and extract the defined features. For traditional LTR algorithms, we use RankLib. We set the number of trees = 1000, the number of leaves = 10, the number of threshold candidates = 256, and the learning rate = 0.1 for MART. We set the iteration = 300, the number of threshold candidates = 10 for RankBoost. We set tolerance = 0.002 and max consecutive selection count = 5 for AdaRank. We set number of bags = 300, number of trees in each bag = 1 and number of leaves for each tree = 100 for Random Forests.

For our proposed approach (EF&ER), we generate

226 single rankers by implementing different parameter and configuration settings in terms of the above four baseline rankers. For instance, we set number of bags = 100, 200, 300 respectively, number of trees in each bag = 1 and number of leaves for each tree = 20, 50, 100 respectively for Random Forests. By iterating over different parameter settings, we can obtain multiple ranker instances based on Random Forests. Similarly, we can obtain multiple ranker instances in terms of other ranking algorithms.

6.3 Evaluation Metrics Normalized Discounted Cumulative Gain(NDCG@N). The discounted cumulative gain (DCG@N) is given by $DCG[n] = \left\{ \begin{array}{l} rel_n \quad if\, n = 1 \\ DCG[n-1] + \frac{rel_n}{log_2 n}, \quad if\, n >= 2 \end{array} \right.$ where rel_n denotes the ranking relevance of the n^{th} community, defined in **Definition 2**. 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) measures the overall ranking accuracy. 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 both $y_i < y_j$ and $f_i < f_j$. Also, $< i, j >$ 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 Tau = $\frac{\#_{conc} - \#_{disc}}{\#_{conc} + \#_{disc}}$.

F-measure@N. F-measure@N incorporates both precision and recall in a single metric by taking their harmonic mean: $F@N = \frac{2 \times Precision@N \times Recall@N}{Precision@N + Recall@N}$. 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 Precision@$N = \frac{|E_N \bigcap E_{\geq 3}|}{N}$ and Recall@$N = \frac{|E_N \bigcap E_{\geq 3}|}{|E_{\geq 3}|}$, where $E_{\geq 3}$ are the communities whose ratings are greater or equal to three.

6.4 Overall Performances We report the performance comparison of our method comparing to six baselines in terms of Tau, NDCG@N and F-measure@N, as shown in Table 4. In all cases we observe a significant improvement with respect to baselines.

Evaluating feature effectiveness. Table 4 shows we control the chosen ranking model and investigate the effectiveness of different feature sets. Specifically, we choose the proposed ensemble ranker as a controlled ranker to study explicit + latent features (EL&ER), latent features (LF&ER), and explicit features (EF&ER). As can be seen, latent features (spatial structure) outperforms explicit features in terms of all Fmeasure@Ns, particularly when N increases. For NDCG@N, latent features perform much better than explicit features from NDCG@5 to NDCG@10 yet slightly worse from NDCG@10 to NDCG@15. This observation shows that the latent features of spatial structure is discriminative for spotting top vibrant communities. A potential interpretation of this observation is that in highly-vibrant communities, the spatial structure including geographic distance and mobility connectivity across POIs are more important for developing vibrant communities, comparing with the number of POIs. Finally, combining both explicit and latent features performs the best, particularly when N is getting larger.

Evaluating model effectiveness. Similarly, we control the chosen feature set and investigate the effectiveness of different ranking models. Specifically, we choose the combination of explicit and latent features as a controlled feature set to compare our ensemble ranking method with AdaRank, MART and RandomForests. Figure 5 shows that the our ensemble geographic ranking method performs the best in terms of NDCG@Ns. This observation indicates the superiority of aggregating multiple rankers to combine all strength of rankers.

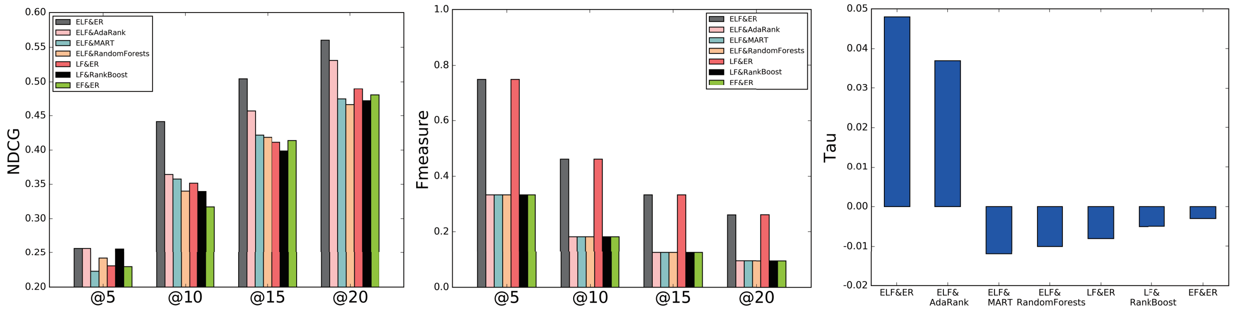
6.5 Robustness Check We use the mean square deviation of NDCG@Ns and Tau to quantify the robustness of our method and baseline methods. The lower deviation, the higher robustness.

Table 5 shows our method (EL&ER) achieves the lowest mean square deviation in terms of NDCG@Ns and Tau. The observation shows that our method is the robustest among the competitors. Another observation which stands out is that when N arises in terms of NDCG@N, our method is still the most robust even though the variance of performances is getting slightly higher. This can be explained as: even though single rankers can generate scores for communities, the final ensemble results may vary due to random initialization. When we ensemble hundreds of ranking score lists into one perspective, the amount of rankers reduces the deviation caused by single ranker. There are only 226 rankers trained to ensemble in the experiment. If the number of rankers increases to 500 even more than $1,000$, the deviation of results could be much lower.

7 Related Work

Related work can be grouped into four aspects, including spatial data mining, learning to rank, model ensemble method and representation learning.

Spatial data mining is the application to use geographical or spatial information to produce business intelligence or other results. The most common spatial information in daily life is the geographical building, POIs



(a) NDCG@N (b) Fmeasure@N (c) Tau

Figure 5: Overall Comparison of Methods

Table 4: Overall Comparison

Method	NDCG @5	NDCG @10	NDCG @15	NDCG @20	Fmeasure @5	Fmeasure @10	Fmeasure @15	Fmeasure @5	Tau
ELF&ER	0.257	**0.441**	**0.504**	**0.560**	**0.75**	**0.462**	**0.333**	**0.261**	**0.048**
ELF&AdaRank	0.257	0.364	0.457	0.531	0.333	0.182	0.125	0.095	0.037
ELF&MART	0.223	0.357	0.422	0.474	0.333	0.182	0.125	0.095	-0.012
ELF&RandomForests	0.242	0.340	0.419	0.466	0.333	0.182	0.125	0.095	-0.010
LF&ER	**0.273**	0.351	0.411	0.489	0.75	0.462	0.333	0.261	-0.008
LF&RankBoost	0.256	0.340	0.399	0.472	0.333	0.182	0.125	0.095	-0.005
EF&ER	0.229	0.317	0.414	0.480	0.333	0.182	0.125	0.095	-0.003

Table 5: Robustness Comparison

Method	NDCG @5	NDCG @10	NDCG @15	NDCG @20	Tau
ELF&ER	**0.052**	**0.133**	**0.262**	**0.358**	**0.073**
ELF&AdaRank	0.165	0.391	0.502	0.693	0.133
ELF&MART	0.090	0.284	0.437	0.766	0.164
ELF&RandomForests	0.289	0.376	0.589	0.822	0.098

and GPS trajectory data. For example, [12] developed a geographical function ranking method by incorporating the functional diversity of communities into real estate appraisal. [11] ranked estates based on investment values by mining users' opinions about estates from online user reviews and offline moving behaviors. [13] and [14] proposed a geographic method, named ClusRanking, for estate appraisal by leveraging the mutual enforcement of ranking and clustering power. [21] and [20] developed a joint model that integrates Mixture of Hawkes Process (MHP) with a hierarchical topic model to capture the arrival sequences with mixed trip purposes.

Also, our work is related to Learning-To-Rank method, which includes pointwise, pairwise, and listwise approaches. The pointwise methods [15] 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 [4, 8]. The listwise methods optimize a ranking loss metric over lists instead of document pairs. For instance, J. Xu et al. proposed AdaRank [24]. More recent work [17] further learns the ranking model which is constrained to be with only a few nonzero coefficients using L1 constraint and propose a learning algorithm from the primal dual perspective.

Moreover, our work can be categorized into model ensembling methods. Paper [6] proposes a bayesian voting based ensembling method in which the ensembling consists of all of the hypotheses in $\mathcal{H}$, each weighted by its posterior probability $P(h|S)$, where S is the training sample. The result is voted by the posterior probability "committee". Paper [3] ran learning algorithm several times and each time constructing the training sets by leaving out disjoint subsets of training data. Freund and Schapire [9] developed the AdaBoost method, maintaining a set of weights for data points and increasing weights of the incorrectly classified examples.

Besides, our work has connection with representation learning. Bengio et al. [1] used Restricted Boltzmann Machines (RBMs) to perform unsupervised feature learning for natural image modeling. [5, 23] explored kernal methods in nonlinear learning and proposed the first analysis of Random Binning(RB) from the perspective of optimization, which by interpreting RB as a randomized block. Paper [18] introduced "t-SNE" built on this geometric perspective adopting a non-parametric approach, based on a training set nearest neighbor graph. [25] proposed a new embedding framework, namely "Deep alIgned autoencoder based eMbEdding" (DIME), to deal with the sparse structure in the emerging networks, by introducing the multiple aligned attributed heterogeneous social network concept to model the network structure.

8 Conclusions

This paper presents a geographic learning approach for mimicking and spotting highly-vibrant communities with human mobility and point-of-interests data. We designed a graph embedding based method to learn the latent representations of community spatial structures, and then devised a geographic ensemble ranking model for spotting vibrant communities. We construct a distance graph and a connectivity graph for each community, and embed both graphs into a latent representation of community structure. We leverage a non-negative matrix factorization-based ensemble strategy to identify vibrant communities using ideas from representation learning. Finally, extensive experimental results demonstrate the effectiveness of the proposed method.

9 Acknowledgments

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

References

[1] Y. Bengio, A. C. Courville, and J. S. Bergstra. Unsupervised models of images by spike-and-slab rbms. In *ICML*, pages 1145–1152, 2011.

[2] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[3] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[5] J. Chen, L. Wu, K. Audhkhasi, B. Kingsbury, and B. Ramabhadrari. Efficient one-vs-one kernel ridge regression for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2454–2458. IEEE, 2016.

[6] N. Cristianini and J. Shawe-Taylor. Bayesian voting schemes and large margin classifiers. *Advances in Kernel Methods*, pages 55–68, 1999.

[7] C. H. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE TPAMI*, 32(1):45–55, 2010.

[8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4(Nov):933–969, 2003.

[9] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[10] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 2001.

[11] Y. Fu, Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and J. Yuan. Sparse real estate ranking with online user reviews and offline moving behaviors. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 120–129. IEEE, 2014.

[12] Y. Fu, G. Liu, S. Papadimitriou, H. Xiong, Y. Ge, H. Zhu, and C. Zhu. Real estate ranking via mixed land-use latent models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 299–308. ACM, 2015.

[13] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z.-H. Zhou. Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering. In *KDD*, pages 1047–1056, 2014.

[14] Y. Fu, H. Xiong, Y. Ge, Y. Zheng, Z. Yao, and Z.-H. Zhou. Modeling of geographic dependencies for real estate ranking. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(1):11, 2016.

[15] L. Hang. A short introduction to learning to rank. *IEICE Transactions on Information and Systems*, 94(10):1854–1862, 2011.

[16] L. Jiang. Learning random forests for ranking. *Frontiers of Computer Science in China*, 5(1):79–86, 2011.

[17] H. Lai, Y. Pan, C. Liu, L. Lin, and J. Wu. Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Trans. on Computers*, 62(6):1221–1233, 2013.

[18] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.

[19] E. Malizia and Y. Song. Vibrant downtowns: Can vibrancy explain variations in downtown property performance? *Working Paper. Institute for Economic Development, UNCH*, 2014.

[20] Y. F. H. X. Pengfei Wang, Guannan Liu and Y. Zhou. Spotting trip purposes from taxi trajectories: A general probabilistic model. In *ACM Transactions on Intelligent Systems and Technology*. ACM, 2018.

[21] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal. Human mobility synchronization and trip purpose detection with mixture of hawkes processes. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 495–503. ACM, 2017.

[22] F. Wu and A. G.-O. Yeh. Urban spatial structure in a transitional economy: the case of guangzhou, china. *Journal of the American Planning Association*, 65(4):377–394, 1999.

[23] L. Wu, I. E. Yen, J. Chen, and R. Yan. Revisiting random binning features: Fast convergence and strong parallelizability. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1265–1274. ACM, 2016.

[24] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR*, pages 391–398, 2007.

[25] J. Zhang, C. Xia, C. Zhang, L. Cui, Y. Fu, and P. S. Yu. Bl-mne: Emerging heterogeneous social network embedding through broad learning with aligned autoencoder. In *Proceedings of the 2017 IEEE International Conference on Data Mining*, ICDM '17, 2017.