Spring 2022

Representation learning on heterogeneous spatiotemporal networks

Dakshak Keerthi Chandra

Department: Computer Science

REPRESENTATION LEARNING ON HETEROGENEOUS SPATIOTEMPORAL NETWORKS

by

DAKSHAK KEERTHI CHANDRA

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2022

Approved by:

Dr. Jennifer Leopold, Advisor
Dr. Bruce McMillin
Dr. Chaman L . Sabharwal
Dr. A. Ricardo Morales
Dr. Casey Canfield

ABSTRACT

The problem of learning latent representations of heterogeneous networks with spatial and temporal attributes has been gaining traction in recent years, given its myriad of real-world applications. Most systems with applications in the field of transportation, urban economics, medical information, online e-commerce, etc., handle big data that can be structured into Spatiotemporal Heterogeneous Networks (SHNs), thereby making efficient analysis of these networks extremely vital. In recent years, representation learning models have proven to be quite efficient in capturing effective lower-dimensional representations of data. But, capturing efficient representations of SHNs continues to pose a challenge for the following reasons: (i) Spatiotemporal data that is structured as SHN encapsulate complex spatial and temporal relationships that exist among real-world objects, rendering traditional feature engineering approaches inefficient and compute-intensive; (ii) Due to the unique nature of the SHNs, existing representation learning techniques cannot be directly adopted to capture their representations.

To address the problem of learning representations of SHNs, four novel frameworks that focus on their unique spatial and temporal characteristics are introduced: (i) collective representation learning, which focuses on quantifying the importance of each latent feature using Laplacian scores; (ii) modality aware representation learning, which learns from the complex user mobility pattern; (iii) distributed representation learning, which focuses on learning human mobility patterns by leveraging Natural Language Processing algorithms; and (iv) representation learning with node sense disambiguation, which learns contrastive senses of nodes in SHNs. The developed frameworks can help us capture higher-order spatial and temporal interactions of real-world SHNs. Through data-driven simulations, machine learning and deep learning models trained on the representations learned from the developed frameworks are proven to be much more efficient and effective.

ACKNOWLEDGMENTS

I would like to thank my academic advisors Dr. Jennifer Leopold for her vision and guidance on which problems to research and the academic rigor needed to solve them; my committee members, Dr. Bruce McMillin, Dr. Chaman L . Sabharwal, Dr. A. Ricardo Morales, and Dr. Casey Canfield for their perceptive suggestions during my study and research which helped me to widen my research from various perspectives.

Beyond academics, I would like to thank my family for providing unconditional love, support, and continuous encouragement throughout my Ph.D. This accomplishment would not have been possible without them.

Also, I would like to thank Pradeep J.K. for his guidance throughout this arduous journey; Dr. A. Ricardo Morales for being a positive force during difficult times; Dr. Yanjie Fu for his vision and keen insights; my friends for providing companionship, joyful memories, and for their endless motivation and support.

TABLE OF CONTENTS

Page

SECTION

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

1. INTRODUCTION

Complex real-world systems handle big data that consist of spatial and temporal attributes that lead to an inherently spatial-temporal and interconnected composition that can be structured into Spatiotemporal Heterogeneous Networks (SHNs), thereby making efficient analysis of these networks extremely vital. Most real-world networks as shown in Figure 1.1 are heterogeneous and incorporate underlying spatiotemporal dependencies.

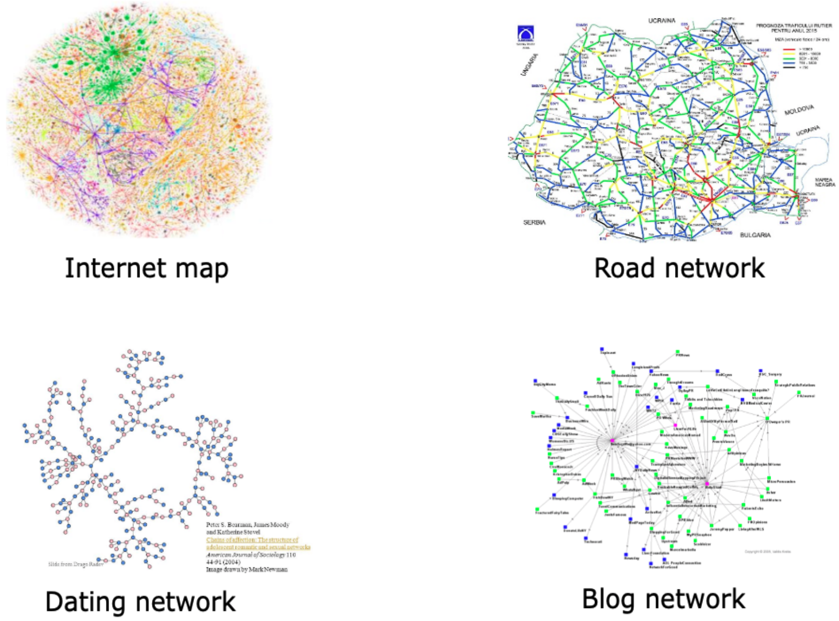

Internet map
Road network
Peter S. Bearman, James Moody
and Katherine Stovel
Chains of affection: The structure of
adolescent romantic and sexual networks
American Journal of Sociology 110
44-91 (2004)
Image drawn by Mark Newman
Slide from Drago Radev
Dating network
Blog network

Figure 1.1. Real-world Spatiotemporal Heterogeneous Networks

In today's digital world, spatiotemporal data has become increasingly available. Mining spatiotemporal data can help us understand the intrinsic structure of systems while simultaneously revealing the dichotomy of multi-class spatial and temporal relationships that exist between objects. Effective analysis of spatiotemporal data can help us understand the semantics of real-world systems, identify trends, forecast future events, and detect anomalies.

For example, in a vehicle with an autonomous driving system, sensors are connected to monitor the condition of the terrain. Understanding underlying temporal and spatial structure is imperative for the system to develop appropriate accident mitigation measures. Smart transportation systems with advanced spatial and temporal information management capabilities can help monitor traffic flow and safety, find the optimal route to get from point A to point B (Route Planning), etc. By learning Spatiotemporal relationships we can better quantitatively depict urban regions and understand factors influencing rapid growth, expansion, and changes. SHNs also have applications in the field of healthcare, agriculture, etc.

1.1. REPRESENTATION LEARNING

In recent years, representation learning models have proven to be quite effective in capturing lower-dimensional representations of the data that can reduce variance and support effective machine learning. The performance of machine learning frameworks is heavily dependent on the choice of data representation (or features) on which they are applied. For that reason, much of the actual effort in deploying machine learning algorithms goes into the design of preprocessing pipelines and data transformations that result in a representation of the data that can support effective machine learning. Such feature engineering is important but labor-intensive and highlights the weakness of current machine learning algorithms, their inability to extract and organize the discriminative information from the data. Feature engineering is a way to take advantage of human ingenuity and prior knowledge to compensate for that weakness. To expand the scope and ease of applicability of machine learning, it would be highly desirable to make machine learning algorithms less dependent on feature engineering so that novel applications could be constructed faster, and more importantly, to make progress towards Artificial Intelligence (AI). An AI must fundamentally understand the world around us, and we argue that this can only be achieved if it can learn to identify and disentangle the underlying explanatory factors hidden in the observed data.

Real-world data are complex, heterogeneous, and high dimensional in nature. The curse of dimensionality is one of the major problems that these models have to address. Many machine learning and deep learning models struggle when dealing with high dimensional data. Operating on high dimensional data is computationally expensive both in terms of time and space. Many of the features in the data can be highly correlated and carry little to no statistical significance. Contrarily, these features can make the model overfit and affect its interpretability. Hence, it is necessary to reduce the data dimensionality and select the most important features. Learning representations of the data that make it easier to extract useful information when building classifiers or other predictors. In the case of probabilistic models, a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input. A good representation is also one that is useful as input to a supervised predictor. Among the various ways of learning representations, this dissertation focuses on deep learning methods, those that are formed by the composition of multiple non-linear transformations, with the goal of yielding more abstract and ultimately more useful representations. The rapid increase in scientific activity on representation learning has been accompanied and nourished by a remarkable string of empirical successes both in academia and in industry. Representation learning has a strong impact in the area of network embedding, speech recognition, and signal processing, object recognition, natural language processing, multi-task and transfer learning, etc. with breakthrough results.

1.1.1. Representation Learning on Networks. Networks are one of the most powerful structures for modeling problems in the real world. Machine learning models defined on learned representations/embedding of complex real-world networks have the potential to solve a variety of problems. Models developed for learning representations of networks map graph nodes to vectors of real numbers in a multidimensional space. To be useful, a good embedding should preserve the structure of the graph. The vectors can then be used as input to various network and graph analysis tasks. With link prediction, for instance, one can predict whether two persons will become friends on a social network. Many machine learning algorithms, however, require that each input example is a real vector. Representation learning models for a network encompass various methods for unsupervised, and sometimes supervised, learning of feature representations of nodes and links in a network. Typically, network representation learning methods are based on the assumption that the similarity between nodes in the network should be reflected in the learned feature representations.

1.1.2. Representation Learning on Spatiotemporal Networks. Spatiotemporal networks are spatial networks whose topology and parameters change with time. We aim to investigate how these additional dimensions influence the structural properties and the dynamic behavior of networks. From the spatial point of view, we study how the nodes of a network can be placed in a metric space and how distance affects the pattern of connections among them. These networks are important due to many critical applications such as emergency traffic planning and route-finding services and there is an immediate need for models that support the design of efficient algorithms for computing the frequent queries on such networks. This problem is challenging due to potentially conflicting requirements of model simplicity and support for efficient algorithms.

Large amounts of spatiotemporal data are collected every day from several domains, including georeferenced climate variables, epidemic outbreaks, crime events, social media, traffic, and transportation dynamics, among many others. Analyzing and mining such kinds of data is of great importance for advancing the state-of-the-art in many scientific problems and real applications. Nevertheless, data with spatial and temporal characteristics have different properties in comparison to relational sources studied in classical data mining literature. They present temporal and/or spatial dependencies, in which instances are not independent or identically distributed. It means that samples can be structurally related in some spatial regions or specific temporal moments. Also, they are non-static, i.e., the instances can change their class attribute depending on time and location. Thus, traditional data mining methods are not the ideal tools for spatiotemporal data, which can result in poor performance and misleading interpretation.

1.2. DISSERTATION STATEMENT

To facilitate efficient analysis of spatiotemporal data and in-turn Spatiotemporal Heterogeneous Networks through representation learning this dissertation develops general frameworks that can be used to create SHNs that can accurately represent spatiotemporal data and learn effective representations from the constructed SHNs. Specifically, the dissertation introduces four novel frameworks that adopt the following approaches:

- Collective representation learning framework with features importance: We develop a graph-based framework that collectively integrates the multi-modal spatiotemporal relationships through representation learning.

- Modality aware representation learning framework: Spatial-temporal data consists of multiple dimensionalities. Hence, in this framework, we develop a tensor factorization-based approach to integrate multi-class relationships and learn their representation through tensor factorization.

- Distributed representation learning framework for learning human mobility patterns: We develop a multi-graph-based framework that leverages power-law distribution associated with human mobility patterns to learn representations.

- Representation learning framework with node sense disambiguation: The state of relationships between nodes in an SHN is everchanging with respect to spatial and temporal contexts. Thus we develop a framework that captures the representations of such relationships between nodes by learning their contrastive contextual node senses.

1.3. DISSERTATION ORGANIZATION

In Section 2, literature review of some of the previous related work is provided to demonstrate the idea of information flow. It briefly discusses research work related to the developed frameworks. In Section 3, we study the problem of learning representations of urban communities by investigating the structural behavior patterns of users. Section 4 presents the modality aware representation learning framework that is based on user profiling by investigating the mobility patterns. Section 5 shows how the interactions between human and the physical environment is modeled to learn representations by leveraging NLP based algorithms. Section 6 presents the framework that models the dynamic nature of spatio-temporal relationships between nodes as a contrastive node sense learning problem and learn representations. Section 7, concludes the dissertation with a discussion on potential future research directions.

2. LITERATURE REVIEW

In this section, a literature review of some of the previous work is introduced. This review helps in demonstrating the idea of information flow. We briefly discuss research work related to the study of urban informatics, representation learning for network embedding, learning word embeddings in NLP, and recommender systems. The section on urban informatics consists of research work that studies experiences of people in the context of cities and urban environments. The second section includes research work related to the use of representation learning for network embedding, and the third section dicusses research work pertaining to learning word embeddings in NLP. The fourth section consists of work that have been widely used for building state-of-the-art recommender systems used for recommending relevant items to users. In addition to the above categories, in the fifth section we also discuss other research work specifically related to some of the techniques adapted by the frameworks introduced in the dissertation.

2.1. URBAN INFORMATICS

Our work is partly related to urban informatics. Urban informatics focuses on learning urban community structure by studying intricate spatial and temporal relations created from dynamic human mobility between static geospatial entities of a city. With the increasing availability of massive amounts of GPS and check-in data from taxi services and social media, we can study the multiplex of overlapping factors that influence the growth of urban environments in cities. Urban informatics uses geographical or spatial information to produce business intelligence or other results. The most common spatial information in daily life is the geographical building, POIs, and GPS trajectory data. In [1], Fu et al. developed a geographical function ranking method by incorporating the functional diversity of communities into real estate appraisal. The work in [2] ranked estates based on investment values by mining user opinions about estates from online user reviews and offline moving behaviors. [3] and [4] developed a geographic method, named ClusRanking, for estate appraisal by leveraging the mutual enforcement of ranking and clustering power. The network embeddings learned from our models based on the constructed SHINs consist of rich semantic and mobility information, which can help us understand the multiplex of overlapping factors that propel the growth of a city.

2.2. REPRESENTATION LEARNING ON NETWORK EMBEDDING

Representation learning for network embedding is primarily used to find a way to represent, or encode, network structure. It aims at learning low-dimensional representations for the vertices of a network such that the proximity among them in the original space is preserved. Machine learning models can then easily exploit the learned embeddings to perform various tasks. Here we review some key advancements of network embedding in the network domain. One such successful method that tries to learn the latent structure of a network is random walk [5]. [6] develops to learn the network embeddings based on random walk statistics. Thus, instead of using a deterministic measure of graph proximity, unlike methods [6, 7] these random walk methods employ a flexible, stochastic measure of graph proximity, which has led to superior performance in a number of settings [8]. [9] and [10] are two such approaches that use random walks. Our work is influenced by DeepWalk [9]. DeepWalk combines random walk proximity with the skip-gram model [11], a language model that maximizes the co-occurrence probability among the words that appear within a window in a sentence. Node2vec [10] is another method that utilizes random walks. It develops a biased random walk procedure to explore the neighborhood of a node. GraRep [12] is a method that learns a latent representation of vertices on graphs to capture the global structural information. LINE [13] is a method that can deal with arbitrary types of information networks: undirected, directed, and/or weighted.

Meta-path [14] is another such method that is used to calculate the proximity between two vertices in a network. PathCount is a method that measures the number of meta-path instances connecting the two objects, and PathSim is a normalized version of it [14, 15]. Path constrained random walk (PCRW) [16, 17] is a more sophisticated way to define the proximity between two vertices based on an instance of meta-path $\mathcal{P}$. PCRW calculates the probability that a random walk restricted on a meta-path would follow the instances connecting two objects. PCRW was initially developed [16] for the task of relationship retrieval over bibliographic networks. Later, [18] developed an automatic approach to learn the best combination of meta-paths and their corresponding weights based on PCRW.

However, the above models fail to harness complex dynamics of spatial and temporal attributes that highly influence complex multi-class relationships that exists between nodes of SHNs. Besides, computing the graph embeddings of SHNs based on naive random walks or path constrained random walks for information propagation is inefficient as they lack semantic information. To address these shortcomings and facilitate learning of efficient representations of SHNs this dissertation puts forward four different approaches.

2.3. LEARNING WORD EMBEDDINGS IN NLP

Our work is also partly influenced by advancements made in word embedding techniques used in the NLP domain. Human language is an integral part of intelligence, but it poses some challenges for continuous gradient-based learning algorithms. Discreteness, sparsity, high dimensionality, and sequences having variable length can be problematic for machine learning algorithms, in a way different from other data modalities such as images. This means that special care needs to be taken to learn representations that are useful for this data modality. A very popular approach for NLP and text mining problems has been to learn so-called word embeddings. Several methods have been developed to learn word embeddings from text corpora [11, 19, 20]. Some techniques have been developed to learn embeddings jointly using knowledge base relations and corpus co-occurrences

[21]. Bilingual distributional representations have also attracted a lot of attention, and researchers have addressed the training of bilingual embeddings using several approaches [22, 23]. Due to the availability of large multilingual semantic networks and the need for an accurate representation of word senses in different languages, some work have also been developed for multilingual embedding. [24] is one such method that combines text-based and knowledge-based methods, yielding multilingual vector representations.

2.4. RECOMMENDER SYSTEMS

Recommender systems have been widely used for recommending relevant items to users. They are a subclass of information filtering that seeks to predict the "rating" or "preference" a user would give to an item. Due to the availability of spatiotemporal data involving users' check-in activity, a great deal of work for providing location-based services by recommending POIs to users based on their preference by using relevant information has also been developed. Most of the related work use collaborative filtering [25], content-based filtering [26], or hybrid [27]. Many Learning to Rank (LTR) methods have also been developed for reliably recommending top-N ranked items. LTR methods can be categorized into point-wise, pair-wise, and list-wise methods. Point-wise methods [28] use scores assigned by individual users to items for learning ranking models. Binary classifiers are learned by pair-wise LTR methods [29]; they compare ordered pairs for deciding if the first item is preferred to the second. The pair-wise classifiers can be computationally expensive as they have to generate training samples for binary classifiers. However, the pair-wise approach works better in practice than the point-wise approach as the strategy of predicting relative order is more similar to the nature of ranking than predicting class labels or relevant scores. List-wise methods [30] leverage the entire list of items related to the user to optimize a list-wise ranking function that measures the distance between the reference lists of ranked items in the training data and the ranked list of items produced by the ranking model.

2.5. OTHER RELATED WORK

In this section we discuss some of the work that is closely related to the techniques adapted in the developed frameworks. Chemla et al. [31, 32] developed the static rebalancing problem paper, which deals with balancing Demand/supply of bikes. The paper addressed the problem of redistribution of bikes to different locations. It presents efficient algorithms for solving instances of reasonable size, and contains several theoretical results related to this problem. Faghih-Imani et al. studies the decision process involved in identifying destination locations after picking up a bicycle from a shared-bike station, in the form of a multinomial logit model[33], The paper by Chen et al. [34] developed a dynamic cluster based model to predict over demand of bikes taking into account the common contextual factors opportunistic contextual factors that affect the bike usage pattern. Work by Yang et al. developed a spatio-temporal bicycle mobility model based on historical bike-sharing data, and devised a traffic prediction mechanism on a per-station basis[35]. Zhang et al. introduce a new trip destination prediction and trip duration inference model on the basis of analyzing individuals' bike usage behaviors on traditional bike sharing systems[36]. Singla et al. developed a incentivizing approach for balancing bike sharing systems. The authors develop a model to engage the users themselves to solve the imbalance problem in bike sharing systems by providing them incentives to ride bikes from station suffering from demand shortage of bikes[37]. Liu et al [38] developed a model for bike re-balancing and data optimization. Meng et al. [39] wrote a paper in which they developed a complete methodology for introducing bike-sharing systems. The developed methodology takes into account potential demand for bicycle use and the willingness to pay of future users for faster journey times, and also introduces a location model for fixing the bicycle pick-up and drop-off stations made with the help of a geographical information system. Lin and Yang studied the strategic planning of public bicycle sharing systems while considering the interests of both users and investors, the developed model attempts to determine the number and locations of bike stations[40]. Caggiani et al. developed a flexible fuzzy decision support system for the redistribution process in traditional bicycle sharing systems with the main aim to minimize the redistribution costs for bike-sharing companies, determining the optimal bikes repositioning flows, distribution patterns and time intervals between relocation operations; overall the objective was of a high level for user satisfaction[41]. Froehlich et al. developed a model which adopted a Bayesian network to predict station status based on the current time and current available dock number[42]. Kaltenbrunner et al. [43] developed a short term prediction of the number of available bikes in stations via the analysis of cyclic mobility patterns. It detects temporal and geographic mobility patterns which are applied to predict the number of available bikes for any station. The predictions are used to improve the bicycle program. Li et al. developed a hybrid and hierarchical prediction model to predict the number of bikes that will be rented from/returned to each station cluster in the early future[44]. Espegren et al. wrote a a paper that considers the static bicycle repositioning problem (SBRP), which deals with optimally re-balancing bike sharing systems (BSS) overnight by using service vehicles to move bikes from full stations to empty stations. A new and improved mathematical formulation for the SBRP is developed[45]. It makes fewer assumptions and considers factors such as heterogeneous fleet, multiple visits to each station, and non-perfect re-balancing. The above models choose clustering based approach, decision based approach, incentivizing approach, or static re-balancing approach to answer problems such as balancing demand and supply of bikes, redistribution of bikes. These models tend to ignore the unique nature of stationless bike-sharing systems where the user gets to pick up and drop off a bike at any location. They tend to focus on the static location-based information, ignore the temporal aspect, and suffer from data sparsity problems. In the developed modality aware framework, by modeling the spatiotemporal data as a multimodal tensor we focus on both the spatial and temporal aspects of the data. We also leverage the use of tensor factorization to predict missing data values, which further enhances the ability of the developed model to address the problem of balancing demand and supply of bikes, redistribution of bikes, and identifying regions with parking problems.

Our work is also very closely related to Tensor Factorization. Rendle et al. developed Factorization machines that combines the advantages of Support Vector Machines (SVM) with factorization models[46]. Xiong et al. developed a temporal collaborative filtering with bayesian probabilistic model using tensor factorization[47]. The work by Oentaryo develops a Hierarchical Importance-aware Factorization Machine (HIFM), which provides an effective generic latent factor framework that incorporates importance weights and hierarchical learning[48]. The work by Karatzoglou introduces a Collaborative Filtering method based on Tensor Factorization (TF), with types of context considered as additional dimensions in the representation of the data as a tensor[49].The work by Rendle et al. [50] presents the factorization model PITF (Pairwise Interaction Tensor Factorization) which is a special case of the tensor decomposition model with linear runtime both for learning and prediction.

3. COLLECTIVE REPRESENTATION LEARNING FRAMEWORK WITH FEATURES IMPORTANCE

In this section, a collective embedding framework that leverages the use of auto-encoders and Laplacian score to learn effective embeddings of spatiotemporal networks of urban communities is presented. In addition, it also introduces a novel weighted degree centrality measure for constructing spatiotemporal heterogeneous networks. To evaluate the performance of our developed model, it is tested on real-world urban community data. Experimental results demonstrate the effectiveness of our model over state-of-the-art alternatives.

3.1. BACKGROUND AND OVERVIEW

Many machine learning and deep learning models struggle when dealing with high dimensional data. The curse of dimensionality is one of the major problems that these models have to address when dealing with real-world data as they are heterogeneous. Operating on high dimension data is computationally expensive both in terms of time and space. Many of the features in the data can be highly correlated and carry little to no statistical significance. Contrarily, these features can make the model overfit and affect its interpretability. Hence, it is necessary to reduce the data dimensionality and select the most important features.

Consequently, many machine learning models use filter or wrapper based feature elimination methods as a way to deal with this problem. Filter methods use some mathematical evaluation function that is based on intrinsic characteristics of the features, like correlation or mutual information. Wrapper based methods adopt more of a greedy search approach by evaluating all the possible combinations of features against the evaluation criteria. In recent years, feature extraction methods or representation learning models have also proven to be quite effective in addressing the high dimensionality problem as they are capable of capturing lower-dimensional representations of the data that can reduce variance and support effective machine learning. However, most of these models are ineffective when dealing with high-dimensional spatiotemporal data. Models using filter or wrapper-based techniques tend to be compute-intensive. These models also lack an efficient approach to quantify the interconnectedness strength of nodes in SHNs. Hence, they struggle to extract and organize the discriminative information from the complex spatial and temporal relationships that exist between real-world objects.

Learning efficient embeddings of a spatiotemporal heterogeneous network (SHN) can be quite challenging due to the complex nature of the relationships it encompasses. Figure 3.1 presents an example of a region's SHN, where nodes are Point of Interests (POIs) and edges are comprised of sptiotemporal relationships.

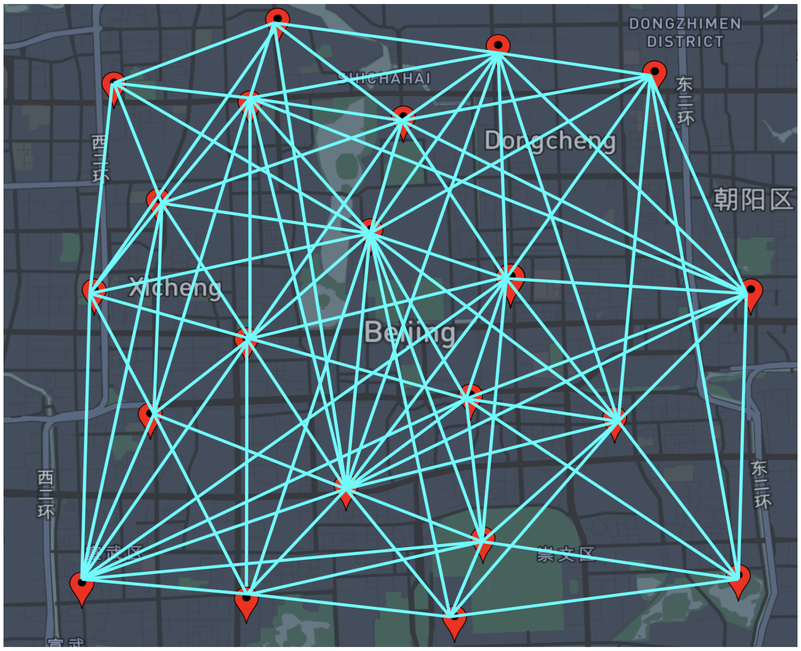


Figure 3.1. SHN depicting human mobility between POIs

We have to consider the static spatial relationships that pertain to geographical structures and take into account the dynamic mobility patterns that exist between geographical structures. By analyzing spatial relationships we can learn about the spatial allocations and the significance of geographical structures. By studying mobility patterns we can learn about the dynamic nature of the relationships between geographical structures. Effective analysis of these factors can help us better understand the evolution of cities and boost commercial activities. All the above overlapping factors suggest the need to study how to quantify and discover urban community structures. However, due to the heterogeneous and complex nature of urban communities, learning urban community structure can be challenging. Three unique challenges that arise in achieving this goal are:

- Efficient representation of spatial and temporal aspects of an urban community as a set of SHNs.
- Collectively learning the embeddings of geographical structures from multiple SHNs.
- Mapping Point Of Interest (POI) Embeddings based on feature importance for learning effective representations of urban communities.

In what follows, we outline how we develop a unified approach for tackling these challenges. Influenced by representation learning, we develop a deep autoencoder based collective embedding framework to learn the embeddings of urban SHNs. We approach the problem of learning urban community structure as an SHN representation learning task. The developed framework can collectively learn the embedding of community structures from spatiotemporal autocorrelation among multiple SHNs by combining static geographical structures data with dynamic human mobility data as periodic spatiotemporal heterogeneous networks.

We first construct a set of SHNs using a new weighted degree centrality measure. We then utilize the developed collective embedding framework for learning the embeddings of important geographical structures or points of interest (POI) from multiple SHNs. We also leverage the use of Laplacian scores [51] to ascertain the importance of latent features in the generated embeddings and in turn preserve the intrinsic structure of the network. This approach helps us to effectively regularize the model and enhances generalization capability.

We evaluate the performance of our developed embedding framework on the task of identifying thriving urban communities in a region and by comparing the quality of the learned embeddings against state-of-the-art network embedding methods. Our experimental results show that the developed method outperforms all alternative approaches in most of the qualitative measures used.

Overview. The remainder of this Section 3 is organized as follows. In Section 3.2, we provide the problem definition for a deep autoencoder based collective embedding framework. Section 3.3 gives detailed information about how we quantify thriving communities. Section 3.4 provides details about the framework of our developed model. In Section 3.5, we report the experimental results of our developed model. Section 3.6 provides a brief review of related work. Finally, we conclude in Section 3.7.

3.2. PROBLEM STATEMENT

This section provides some key definitions to formulate the problem of learning the embeddings of urban SHNs and also give a brief overview of the developed framework. The following definitions are initially stated to help break down the problem and then formalize the community learning problem.

Definition 1: *(Urban Community):* An urban community c_k is made up of multiple residential and commercial buildings. It also consists of many geographical structures (POIs) that provide a variety of urban functions and living services to residents of the community.

Definition 2: *(Spatiotemporal Heterogeneous Networks):* SHNs encapsulate the complex spatial and temporal relationships that exists among real-world objects. Here, POIs are considered as vertices and the frequency of human mobility between POIs are weighted links between POIs. It can be represented as a SHN, $S = (P, E)$, where $P = \{p_i\}_{i=1,....,n}$ are POIs as nodes, and $e_{i,j} = \left(p_i, p_j\right) \in E$ is a link indicating human mobility relationship between two vertices.

Definition 3: *(Periodic SHNs):* The human mobility patterns between POIs are captured based on different time intervals. This is done to accurately capture the generalized dynamics of human mobility. For each day of the week seven periodic SHNs are extracted.

Definition 4: *(Community SHN embedding):* A community SHN embedding is a vector representation of the SHN of a community. The vector representation encapsulates the static spatial configuration of POIs as well as the dynamics of human mobility of the respective community.

Problem Definition. Given a set of periodic SHNs $S(k) = \{s_1^k, s_2^k, ...s_7^k\}$. We formulate a function $f(c) = S^k \rightarrow \mathbb{R}^d$ that projects each vertex $v \in V$ to a vector in d such that static and dynamic relationships within the SHN of a community are preserved.

3.3. QUANTIFYING THRIVING COMMUNITIES

Presence of thriving communities in a region can help us understand the evolution of cities and urbanization in general. Efficient representation of spatial and temporal aspects of an urban community as a set of SHN can help us gain valuable insights. A significance of a community can be measured based on the number of people visiting a community, and on the variety of services a community can provide. The number of people visiting a community can be calculated as a density measure based on the POI check-in activity within the region. The variety of services a community provides can be calculated as a diversity measure based on the presence of different categories of POIs and number of POIs in a community. This approach helps us avoid considering densely packed and economically poor regions as thriving communities as they lack in terms of the variety of services a thriving community can provide.

Here, we develop a function to measure the significance of the community based on the diversity and density aspects of a community. Density of a community a community c_k is measured as the sum of number of check-in activities for POIs within a community:

$$den(c_k) = \sum_{i=1}^{n} visits(p_i) \tag{3.1}$$

where, $den(c_k)$ is the density measure of the community c_k and $visits(p_i)$ is the number of visits to the i^{th} POI

Diversity of a community c_k is measured by considering the number of POIs present in the community as well as the number of POI categories:

$$div(c_k) = 1 - \frac{\sum_{i=1}^{n} P_i(P_i - 1)}{P_c(P_c - 1)} \tag{3.2}$$

where, P_i is the total number of POIs in individual POI categories and P_c is the total number of POI categories.

Finally, we measure the significance of a community c_k by fusing density and diversity aspects of the community:

$$Sig(c_k) = den(c_k) \times div(c_k) \tag{3.3}$$

Once the significance scores of all the communities in a region are calculated, the significance scores of individual communities are normalized:

$$Sig(c_k)_{norm} = \frac{Sig(c_k) - Sig(c_k)_{min}}{Sig(c_k)_{max} - Sig(c_k)_{min}} \tag{3.4}$$

We use the developed function to measure the significance score of each community. Based on the significance score, the problem of identifying thriving communities is modeled.

3.4. METHODOLOGY

In this section, we describe our developed model. In Section 3.4.1 we elaborate on the process of constructing SHNs. Then, we introduce our developed collective embedding framework in Section 3.4.2 that is used to generate embeddings from the periodic SHNs.

Finally, in Section 3.4.3 we implement a weighted node centrality based method to systematically align and aggregate POI embeddings for community structure representation learning.

3.4.1. SHN Construction. Due to the heterogeneous nature of large scale SHNs, it is challenging to ascertain the connectivity measurements between POIs of a community. Ideally, to estimate the connectivity between two POIs we would directly count the total number of visits between them. However, real-world data is messy, and we don't always have exact details about the origin POI or destination POI. This is because spatiotemporal data is collected from numerous sources like smartphones, bike pickup and drop location, social media check-ins, etc. For instance, data from Mobike trips might include a start and an endpoint, but they don't always accurately represent the exact origin POI and destination POI. Hence, we use a probabilistic function as an estimation method that fits spatiotemporal data from various sources and does not require exact origin POIs and destination POIs.

We first calculate the possibility of a person visiting a POI using Equation 3.5. The equation adheres to the power law distribution which aligns with the fact that people tend to visit POIs which are at close proximity:

$$P(x) = \frac{\beta_1}{\beta_2} \cdot y \cdot exp\left(1 - \frac{y}{\beta_2}\right) \tag{3.5}$$

where, variable y represents the distance between the original drop-off point d and the destination POI p. β_1 and β_2 are the hyperparameters used to control the shape of the function $P(x)$.

The total number of times a POI p_i is visited by users is calculated by aggregating all the drop off points:

$$\tau(p_i) = \sum_{d \in \mathcal{D}} P(dis(d, p_i)) \tag{3.6}$$

where, $\mathcal{D}$ is a set of bike drop-off points in the community.

We then calculate the connectivity measure between a pair of POIs as weighted links by multiplying the $\tau\left(p_i\right)$ with $\tau\left(p_j\right)$, to describe the possibility of users visiting p_j from p_i. Based on the human mobility we can quantify the connectivity between p_j from p_i. The calculation can be formulated as:

$$\tau_{ij} = \begin{cases} \tau\left(p_i\right) \cdot \tau\left(p_j\right), & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{3.7}$$

Once the weights of the links between vertices (POIs) are calculated, we can find the degree centrality measure of vertices in the SHN. For quantification of an SHN's interconnectedness strength a great variety of centrality measures have been developed [52, 53, 54]. The degree centrality measure (DC) is one of the simplest centrality measures, it uses the number of links between vertices as an indicator of a vertex's interconnectedness. The DC measure only concerns networks based on the presence or absence of a link between vertices. However, this approach, when applied on weighted networks will result in loss of information. An increasing number of studies have been focused on finding appropriate measures for weighted networks [52, 55, 56].

In weighted networks, DC is calculated as the sum of weights of links assigned to vertex and represents vertex strength centrality (SC). This method of calculating weighted node centrality may not be optimal as it ignores the individual link weight imbalance factor. Vertices having the same DC and SC but having varying levels of link weight imbalance will be represented as having equal importance. To capture the complex spatiotemporal relationship that exists between vertices of an SHN we need to overcome this disadvantage. To address this problem we formulate a weighted centrality measure that assigns a calculated degree centrality score while accounting for the link weight imbalance factor:

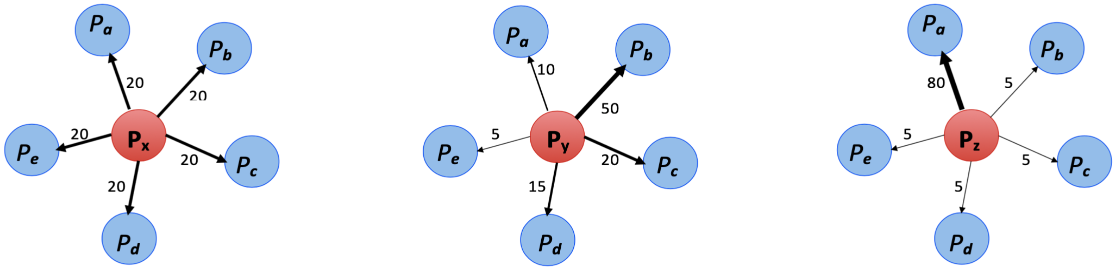
$$WDC_p = \frac{1}{n} \sum_{i=1}^{n} 1 \left\{ p_{wcd_i} \geq \mu \right\} \tag{3.8}$$

where, WDC_p represents weight centrality measure of the vertex. p_{wcd_i} is the weighted cumulative distribution value of link weight p_{w_i}. μ is the mean of $\{p_{wcd_i}, ..., p_{wcd_n}\}$. The weighted cumulative distribution value P_{wcd_i} is calculated as:

$$p_{wcd_i} = \frac{1}{\sum_{i=1}^{n} p_{w_i}} \sum_{i=1}^{n} p_{w_i}, \text{ if } p_{w_i} \leq p_w \tag{3.9}$$

where, P_{w_i} is the link weight instance of connected vertices and P_w is current vertex link weight.

Figure 3.2 shows three scenarios where vertices P_x, P_y, and P_z with varying levels of link weight imbalance have the same DC and SC scores. We can see that the traditional approach for calculating DC and SC does not work here. By using the developed weighted degree centrality measure (WDC_p) we can more accurately quantify the interconnectedness strength of individual vertices in a network.



Example 1	Example 2	Example 3
$DC_{P_x} = 5$	$DC_{P_y} = 5$	$DC_{P_z} = 5$
$SC_{P_x} = 100$	$SC_{P_y} = 100$	$SC_{P_z} = 100$
$WDC_{P_x} = 1$	$WDC_{P_y} = 0.4$	$WDC_{P_z} = 0.2$

Figure 3.2. Three examples where traditional approach of measuring degree and strength centrality does not work.

By using the above steps for accurate representation of SHNs we construct seven periodic SHNs for each day of the week $S(k) = \{s_1^k, s_2^k, ...s_7^k\}$ of community c_k. Here, the SHNs represent the urban community and vertices of SHNs are POIs of the urban community. The SHNs are then fed into the collective embedding framework to capture the static and dynamic relationships of urban communities.

3.4.2. Collective Embedding Framework. In this section, we elaborate on the deep autoencoder based collective embedding framework. Here, we give the detailed architectural information about the collective embedding framework for learning community structure from periodic SHNs. The architecture of the developed framework is shown in Figure 3.3.

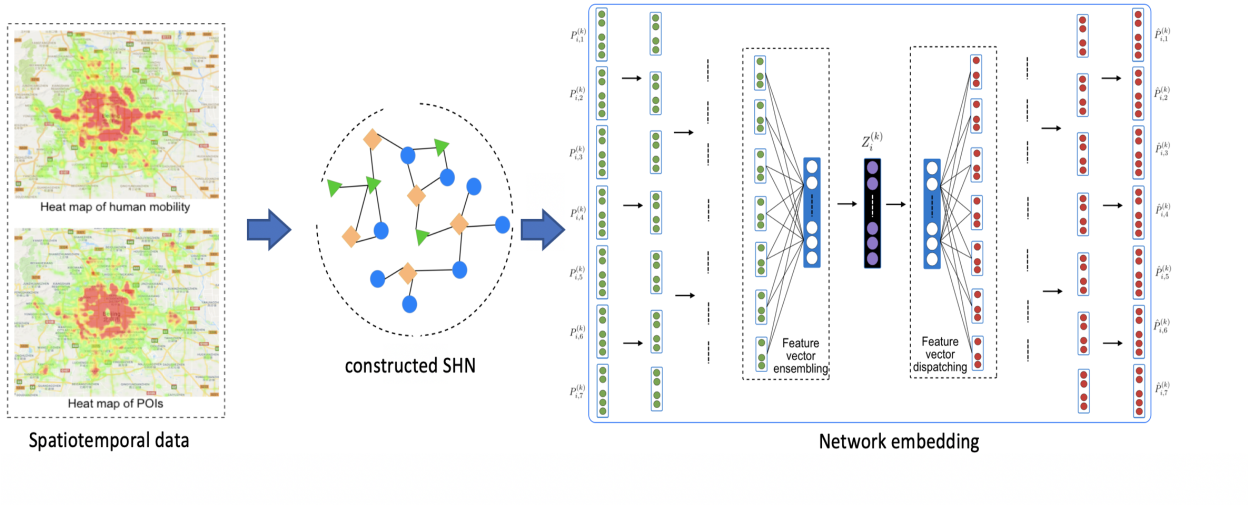


Figure 3.3. Collective embedding framework

The general idea behind an autoencoder is to learn the nonlinear relationship between data samples via an encoder to a hidden layer. The hidden units are then used as the new features by decoders to reconstruct the data:

$$h_i = \sigma\left(w_i e_i + b_i\right); \hat{e}_i = \sigma\left(w_j h_i + b_j\right) \tag{3.10}$$

where $h_i \in \mathbb{R}^n$ is the hidden representation, and $\hat{e}_i \in \mathbb{R}^d$ is the reconstruction of normalized input. $w_i \in \mathbb{R}^{n \times d}$ and $w_j \in \mathbb{R}^{d \times n}$ are the weight matrices, and bias vectors $b_i \in \mathbb{R}^n$ and $b_j \in \mathbb{R}^n$. σ is an activation function.

A traditional auto-encoder for learning the embeddings of the constructed periodic SHNs is not suitable as it can only take one input during each training iteration. To overcome this, A collective learning method is implemented to learn the embeddings of a community's SHN from the inter-correlations of multiple periodic SHNs that capture the spatiotemporal dynamics of the community structure.

Given a community c_k, the i^{th} row of the constructed periodic SHN $S_d^{(k)}$ represents the i^{th} POI $p_{i,d}^{(k)}$ on the day d of the week. For each POI $p_i^{(k)}$ there are seven vectors $\left\{p_{i,d}^{(k)}\right\}^*$ for each day of the week. These seven vectors are used as inputs for POI p_i. An embedding ensemble method is used after the last layer of the encoder part of the auto-encoder to handle the multiple inputs problem. The encoding step is formulated as:

$$\begin{cases} y_{i,d}^{(k),1} & = \sigma\left(w_{i,d}^{(k),1} p_{i,d}^{(k)} + b_{i,d}^{(k),1}\right) \forall d \in \{1, 2, ..., 7\}, \\ y_i^{(k),o+1} & = \sigma\left(\sum_d w_{i,d}^{(k),o+1} y_{i,d}^{(k),o} + b_d^{(k),o+1}\right), \\ z_i^{(k)} & = \sigma\left(w^{(k),o+2} y_i^{(k),o+1} + b^{(k),o+2}\right) \end{cases} \tag{3.11}$$

where, $z_i^{(k)}$ is the lower dimensional space in which encoding of p_i is stored. w and b denote the weight and bias, respectively.

In the decoding part of the the auto-encoder, we use $z_i^{(k)}$ as the input and the final output is reconstructed as $\hat{p}_i^{(k)}$. We pass $z_i^{(k)}$ as seven latent vectors for each day of the week and then reconstruct embeddings at each hidden layer. The relationship among these vector variables can be denoted as:

$$\begin{cases} \hat{y}_i^{(k),o+1} & = \sigma\left(\hat{w}^{(k),o+2} z_i^{(k)} + \hat{b}^{(k),o+2}\right), \\ \hat{y}_{i,d}^{(k),1} & = \sigma\left(\hat{w}_{i,d}^{(k),1} \hat{y}_{i,d}^{(k)} + b_{i,d}^{(k),1}\right) \forall d \in \{1, 2, ..., 7\}, \\ \hat{p}_i^{(k)} & = \sigma\left(\hat{w}_{i,d}^{(k),1} \hat{y}_{i,d}^{(k),1} + \hat{b}_{i,d}^{(k),1}\right) \end{cases} \tag{3.12}$$

where, $\hat{p}_i^{(k)}$ is the reconstructed vector. w and b denote the weight and bias, respectively.

We then formulate the loss function by aggregating the loss of each day d:

$$\mathcal{L}^{(k)} = \sum_{d \in \{1,..7\}} \sum_{i} \left\| \left(p_{i,d}^{(k)} - \hat{p}_{i,d}^{(k)} \right) \right\|_2^2 \tag{3.13}$$

where, $p_{i,d}^{(k)}$ represents the original POI vector and $\hat{p}_{i,d}^{(k)}$ is the predicted vector.

3.4.3. Mapping POI Embeddings Based on Feature Importance. With the developed framework, we obtain the SHN embedding of POIs. However, our main aim here is to learn the embeddings of urban communities as a whole. As an urban community encompasses multiple POIs, we aim to map the learned SHN embeddings of POIs based on feature importance to obtain the representations of urban communities.

Since the size and structure of SHNs vary over different urban communities we approach the task of learning the embeddings of urban communities by mapping learned individual POI embeddings with their respective POI categories. Then, we align the POI category embeddings to their respective urban community embeddings. We order individual POIs with their respective POI categories as the number of POI categories is fixed in every urban community. This helps us deal with the varying structure of urban communities.

The task of mapping learned individual POI embedding with their respective POI categories can simply be achieved by summing up the learned embedding of all the POIs that belong to a particular POI category. However, such a simple approach can be ineffective when dealing with POI embeddings as we cannot ignore the importance of latent features. Each feature encapsulates different characteristics of a POI that can give us insights about the nature of relationships. To quantify the importance of each latent feature we develop the following approach.

To ascertain the importance of latent features in the learned embeddings we leverage the use of Laplacian score [51]. It is used to measure the similarity between two sample features. It calculates the locality preserving power of a feature which inn turn indicates its significance. It is based on the observation that two training samples belong to the same feature set if they are close to each other:

$$L_r = \frac{\sum_{i,j}\left(f_{ri} - f_{rj}\right)^2 S_{ij}}{Var\left(\mathbf{f}_r\right)} \tag{3.14}$$

where, f_{ri} and f_{rj} are the i^{th} and j^{th} sample of feature r, $Var\left(\boldsymbol{f}_r\right)$ is the estimated variance of the r^{th} feature, and $S_{i,j}$ is the edge weight.

The edge weight $S_{i,j}$ between data points x_i and x_j is calculated as

$$S_{i,j} = e^{-\frac{\left\|x_i - x_j\right\|^2}{t}} \tag{3.15}$$

where, t is number of edges linking x_i and x_j with other data points.

The Laplacian score function aims to identify the feature samples that are more similar. It tends to be small for training samples belonging to the same feature set. The edge weight between a pair of POIs is calculated by measuring similarity between embedding vectors of two POIs using improved sqrt-cosine similarity [57]:

$$ISCsim_{i,j} = \frac{\sum \sqrt{p\tilde{S}^{(k)}\left[i,r\right] \times \tilde{S}^{(k)}\left[j,r\right]}}{\sqrt{\sum p\tilde{S}^{(k)}\left[i,r\right]^2} \times \sqrt{\sum p\tilde{S}^{(k)}\left[j,r\right]^2}} \tag{3.16}$$

where, $\tilde{S}^{(k)}$ represents the embedding vectors of all POIs in the community c_k, r is the of r^{th} feature in the POI embedding vector.

Given the r^{th} feature in the POI embedding vectors the Laplacian score is calculated as:

$$\mathcal{L}w_r = \frac{\sum_{i,j}\left(\tilde{S}_{r_i}^{(k)} - \tilde{S}_{r_j}^{(k)}\right)^2 ISCsim_{ij}}{Var\left(\tilde{S}_r^{(k)}\right)} \tag{3.17}$$

where $\mathcal{L}w_r$ represents the Laplacian score of the r^{th} feature, $\hat{S}_r^{(k)}$ represents the r_{th} feature of POI embeddings in the community c_k, and $ISCsim_{i,j}$ is the edge weight

Using Equation 3.13, the latent feature weight set $w(k) = \left\{\mathcal{L}w_1^k, \mathcal{L}w_2^k, ...\mathcal{L}w_n^k\right\}$ is obtained.

$$\hat{S}^{(k)}[v,r] = \sum_{p_i \in \Phi_v} \tilde{S}^{(k)}[i,r] \times w_p^k \tag{3.18}$$

where $\hat{S}^{(k)}$ is the POI-category embedding graph for the community c_k, and Φ_v is the v_{th} POI category.

The POI category embeddings are then mapped to community embeddings. Given a community c_k, each row of $\hat{S}^{(k)}$ is aligned into a vector to form community embedding $S^{(k)}$ such that $S^{(k)} = \left(\hat{S}^{(k)}[1,*], \hat{S}^{(k)}[2,*], ..., \hat{S}^{(k)}[v,*]\right)^T$. $S^{(k)}$ is the output of the developed representation learning framework.

3.5. EXPERIMENTAL SECTION

The performances of the developed model is evaluated against baseline models on real-world urban community and human mobility datasets.

3.5.1. Data Description. This section, provides details about the Mobike, POI, and Weibo & Jiepang datasets used for evaluation. Table 3.1 shows the statistics of the datasets mentioned above.

Table 3.1. Datasets Statistics.

Data Sources	**Attributes**	**Statistics**
Mobike Trips	#trip records	3,214,096
	# users	349,693
	Trip start time	
	Trip start location	
	Trip end location	
	Bike id	
	Time period of records	05/10/2017 - 05/24/2017
POIs	Number of POIs	328,668
	# POI categories	20
Weibo & Jiepang	# Check-In	2,020,967
	# users	212,362
	POI Name	
	POI Location	
	POI address	

Mobike, the stationless bike sharing company, released its Beijing city trip dataset in the 2017 Mobike Big Data Challenge. It contains details of 3,214,096 trips. It includes information about the number of users, trip start time, trip endtime, bike id, trip start location, and trip end location. The POI data set for the city of Beijing was acquired from www.dianping.com, which is a commercial review and recommendation website. It contains details about 328,668 POIs divided into 20 different categories like Hospitals, Malls, Restaurant, theaters, etc. The Weibo & Jeipang datasets together include 2,020,967 check-in entries of POIs in Beijing. It contains details like POI name, POI check-in time, POI address, and POI location.

3.5.2. Spatiotemporal Heterogeneous Network Construction. To evaluate the effectiveness of our developed model on very large and complex SHNs, we construct an SHN from Mobike, POI, and Weibo & Jeipang datasets. This section provides information about the steps followed to construct the SHN. We integrated the POI dataset, Weibo & Jeipang dataset with the Mobike dataset to assimilate the rich multi-class human mobility relationships and the geospatial relationships that exist within the POI dataset. To summarize, our SHN is comprised of 1,765,025 vertices and encapsulates multi-class spatial and temporal relationships like visits per day, average commute speed, geographic proximity, POI categories, number of individual user visits, and average visits.

Additionally, we classify a region as an urban community if the region consists of one or more residential complexes and is surrounded by multiple POIs (within the radius of 1km) that provide various services.

3.5.3. Identifying Thriving Communities. We model the problem of identifying thriving communities as a Learning to Rank (LTR) task. Based on the analysis of the significance score we observe that its distribution complies with the power-law distribution. We notice that only a very few communities have a high significance score; whereas, the majority of the communities have scores that are close to or below the mean value of significance score. From the curve, we identify 4 inflection points. These points are used as 4 measures on which we rank the communities, Figure 3.4.

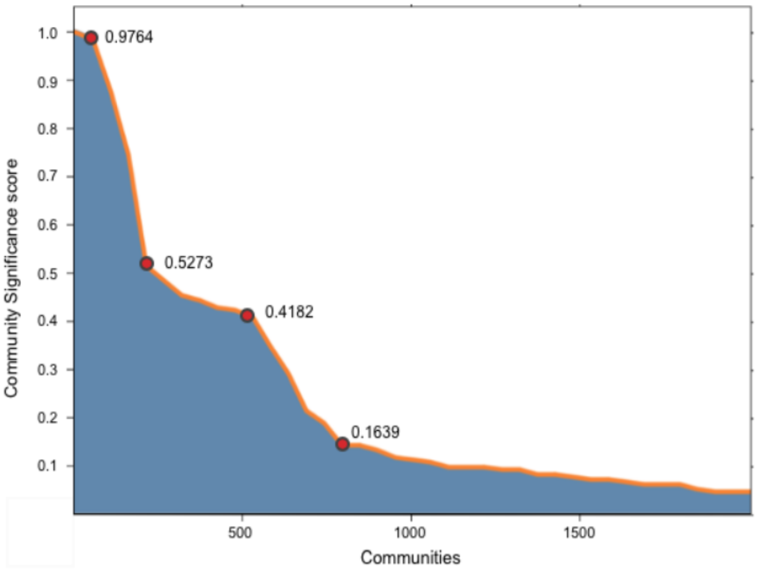


Figure 3.4. Analysis of urban communities

3.5.3.1. Effective feature set selection. For the task of identifying thriving communities and for evaluating the effectiveness of the latent features learned from our model we compare it with explicit features extracted from the data.

- Explicit Features (EF):Features like POI numbers per category, average commute distance, average commute speed, average commute time, number of check-ins, average distance between POIs explicitly pertaining to the data are used.
- Latent Features (LF): Features from the network embeddings generated by the developed collective embedding framework.

3.5.3.2. Baseline models. Application related ranking methods are used to demonstrate the effectiveness of our method.Six Learning to Rank (LTR) methods as baselines are selected for comparison.

- AdaRank (AR) [58]: This is a learning algorithm within the framework of boosting. It works by repeatedly constructing weak rankers based on re-weighted training data. It then linearly combines the weak rankers for ranking prediction. It can minimize a loss function directly defined on the performance measures.

- ListNet (LN) [59]: ListNet is a probabilistic method with two probability models called permutation probability and top one probability that define a listwise loss function for learning.

- Random Forest (RF) [60]: This method works by building an ensemble of decision trees that are trained using the bagging method. It works on the idea that the combination of learning models increases the overall results.

- Multiple Additive Regression Trees (MART)[61]: 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.

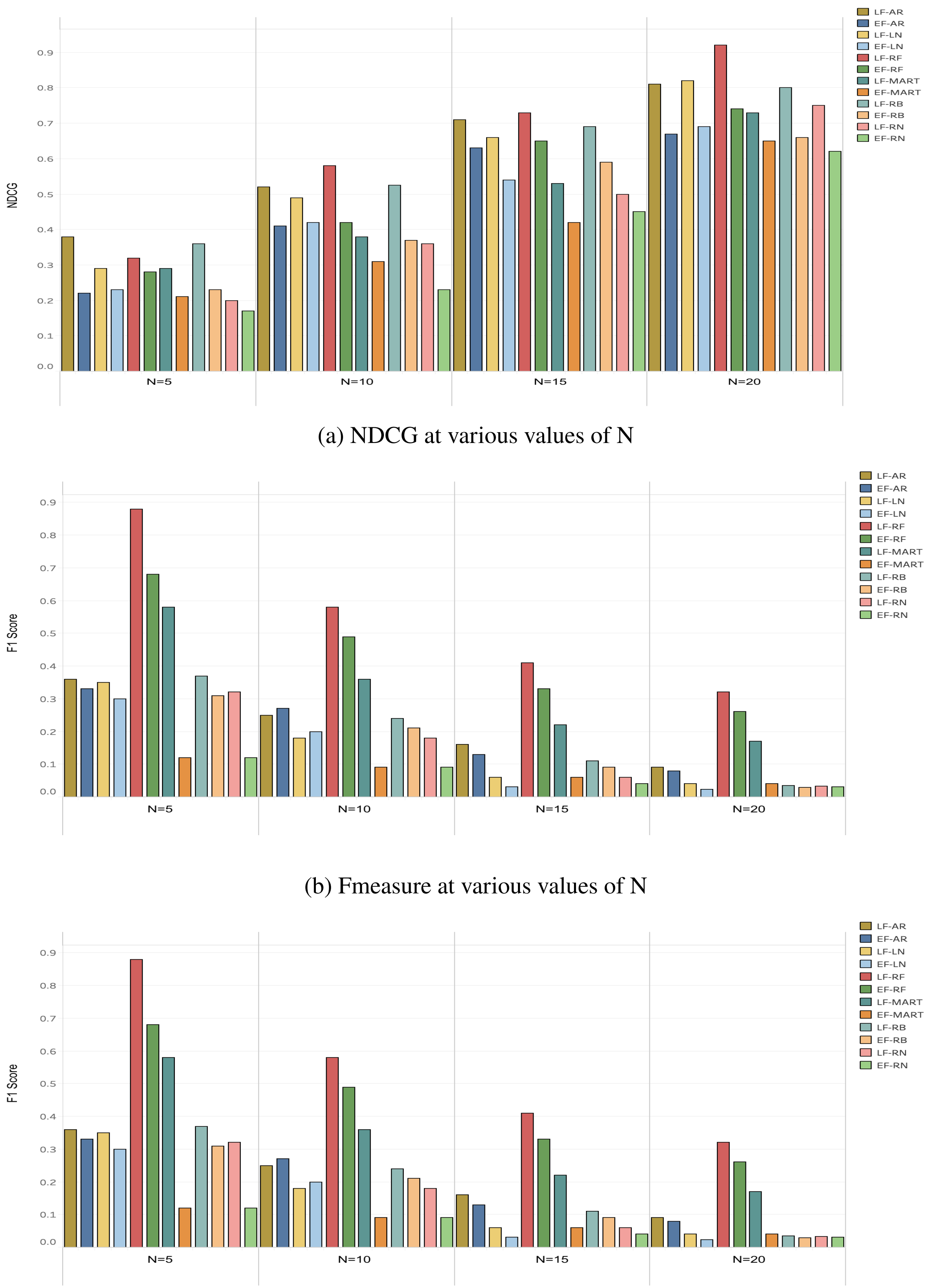
- RankBoost (RB) [62]: RankBoost is a boosted pairwise ranking method, which trains multiple weak rankers and combines their outputs as final ranking.

- RankNet (RN) [63]: RankNet uses a neural network to model the underlying probabilistic cost function.

3.5.3.3. Parameter setting. We use the ranklib2 Python library for implementing the 6 LTR algorithms. For ListNet and RankNet we set learning rate = 0.001, number of hidden layers = 1, the number of hidden nodes per layer = 10, and the number of epochs to train= 500. For RankBoost we set the number of iterations = 500, and number of threshold candidates = 10. For MART we set number of trees = 300, number of leaves = 8, threshold candidtes = 256, and learning rate = 0.1. For Random Forest we set number of bags = 300, sampling rate = 0.5, number of trees = 8, number of leaves = 100. For AdaRank we set number of epochs = 500, tolerence = 0.002, max sampling = 5. We split 75% of dataset into training dataset and the remaining 25% is used as testing dataset. The ranking relevance for communities is based on the significance score $Sig(c_k)$.

3.5.3.4. Evaluation metrics. (1) Normalized Discounted Cumulative Gain (NDCG): This metric is based on the ideal discounted cumulative gain (DCG). NDCG at the n^{th} position is computed as $NDCG\ [n] = \frac{DCG[n]}{DCG'[n]}$.
(2) F1-score: F1-score is a measure of a test's accuracy. It incorporates both precision and recall in a single metric by taking their harmonic mean. $F1 = 2\frac{precision \times recall}{precision + recall}$
(3) Kendall's Tau Coefficient (Tau) : Tau is a measure of rank correlation. We use it to measure 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 a community pair $\langle i, j \rangle$, $\langle i, j \rangle$ 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,$\langle i, j \rangle$ is said to be discordant, if both $y_i < y_j$ and $f_i > f_j$. Tau is given by $Tau = \frac{conc^{-}disc}{conc^{+}disc}$.

3.5.3.5. Results and analysis. Figure 3.5(a) and 3.5(b) show the NDCG and F1-Score of the 6 baseline models for different values of N on both the explicit features extracted directly from the dataset and latent features learned from the developed collective embedding method. Figure 3.5(c) shows the overall ranking accuracy (Tau) of the 6 LTR models on explicit features and latent features. In all baselines models, we can observe significant improvements in performance when trained on latent features. The observations made on the evaluation metrics NDCG, F1-Score, and Tau, demonstrate the superiority of the latent features learned from the developed collective embedding framework. This also shows that the constructed SHNs are capable of efficiently capturing the dynamic and static information of communities. It also shows that our developed weighted degree centrality measure for constructing SHNs can comprehensively and effectively represent static geographical structures as well as dynamic human mobility aspects of a community.



(a) NDCG at various values of N

(b) Fmeasure at various values of N

(c) Tau

Figure 3.5. Overall performance comparisons of the LTR models on latent and explicit features in terms of NDCG, F1-Score, and Tau.

3.5.4. Evaluation of the Learned Embeddings. In this section, we evaluate the quality of embeddings learnt from our developed representation learning framework (CEF-A). We compare it against embeddings learned from other state-of-the-art network embedding methods.

3.5.4.1. Baseline models. We use the below three network embedding methods as baselines for evaluating the efficiency of the learned SHN embeddings.

- LINE [13]: LINE can preserve both first-order and second-order proximities for the undirected network through modeling node co-occurrence probability and node conditional probability.

- GraRep [12]: GraRep preserves node proximities by constructing different k-order transition matrices.

- Node2vec [10]: node2vec develops a biased random walk procedure to explore the neighborhood of a node, which can strike a balance between local properties and global properties of a network.

3.5.4.2. Parameter setting. We use the OpenNE Python toolkit to implement LINE, GraRep, and node2vec models. For node2vec methods, we set the window size = 5, for each vertex we set walk length = 10, and we set the number of walks = 15. For LINE we set negative samples = 5, and the learning rate = 0.025. For GraRep we set maximum matrix transition step K=6.

3.5.4.3. Results and analysis. We use the three baseline models to generate embeddings; the learned embeddings are then passed as inputs into the 6 LTR algorithms to examine the effectiveness of the learned embeddings. We use NDCG as the evaluation metric, and evaluate for different values of N. We then compare the performance of our model against LINE, GraRep, and node2vec models. Table 3.2 shows the experimental results. We can see that our developed framework (CEF-A) outperforms the three baseline models.

Table 3.2. Representation learning evaluation

	Baseline Models	NDCG→N=5	NDCG→N=10	NDCG→N=15	NDCG→N=20
AdaRank	CEF-A	0.467	0.725	0.772	0.805
	LINE	0.112	0.162	0.193	0.445
	GraRep	0.402	0.385	0.227	0.410
	node2vec	0.327	0.612	0.502	0.431
ListNet	CEF-A	0.512	0.783	0.906	0.928
	LINE	0.246	0.105	0.336	0.591
	GraRep	0.397	0.336	0.273	0.412
	node2vec	0.407	0.591	0.441	0.539
MART	CEF-A	0.237	0.421	0.541	0.595
	LINE	0.231	0.310	0.413	0.364
	GraRep	0.164	0.375	0.397	0.412
	node2vec	0.249	0.372	0.451	0.460
Random Forest	CEF-A	0.416	0.487	0.615	0.709
	LINE	0.171	0.408	0.162	0.461
	GraRep	0.342	0.183	0.187	0.438
	node2vec	0.369	0.434	0.426	0.526
RankBoost	CEF-A	0.592	0.683	0.8	0.916
	LINE	0.117	0.409	0.173	0.192
	GraRep	0.351	0.330	0.216	0.234
	node2vec	0.538	0.546	0.398	0.371
RankNet	CEF-A	0.381	0.427	0.75	0.93
	LINE	0.282	0.392	0.497	0.415
	GraRep	0.238	0.364	0.459	0.389
	node2vec	0.147	0.269	0.308	0.406

3.6. CONCLUSION

In Section 3, a novel collective deep autoencoder based embedding framework for efficiently capturing higher-order spatial and temporal interactions of real-world SHNs is introduced. To effectively explore potential spatial and temporal relationships among vertices we employ weighted degree centrality measure which enables us to build spatial-temporal heterogeneous networks that unify and represent static POIs and dynamic human mobility records. To learn the embeddings of POIs from the periodic spatiotemporal mobility graphs, we then develop a deep autoencoder based embedding framework for learning community structure from periodic SHNs. This helps us effectively capture the latent features between POIs in a community. We further develop a mapping procedure that leverages the use of Laplacian score. This approach helps us quantify the importance of each latent feature. It helps us learn community embeddings based on the POI embeddings learned from the developed framework. To evaluate the performance of the developed approach, we applied it to predict thriving urban communities from real-world datasets. We also evaluate the quality of the learned embeddings against state-of-the-art network embedding models. Experimental results show that the embeddings learned from our developed framework can capture the intrinsic structure of the urban community more accurately, and outperforms state-of-the-art alternatives.

4. MODALITY AWARE REPRESENTATION LEARNING FRAMEWORK

In this section, a multidimensional tensor framework which learns from the complex user mobility pattern is presented. It addresses the station-less bike sharing challenges posed by multi-dimension data with spatial and temporal attributes. Specifically, it attempts to resolve some of the problems faced by stationless bike sharing systems. Primary challenges associated are identifying regions within the city suffering from demand shortage of bikes, supply shortage of bikes, and regions with parking problems. The Mobike dataset for the city of Beijing is used to evaluate the presented framework, and the experimental results show the superior performance of the developed framework.

4.1. BACKGROUND AND OVERVIEW

The world has witnessed a rise in popularity of station-less bike sharing systems in recent years, with many cities all over the world implementing them. These station-less bike sharing systems have become especially popular in metropolitan cities like Beijing as they not only help ease the pressure of public transportation systems and help reduce traffic congestion in cities, but also provide an affordable and green way for daily commuters to travel from point A to point B. The station-less bike sharing systems are inherently different from regular bike sharing systems as the bikes are not tied down to stations. With regular bike sharing systems, the commuters need to pick up bikes from a station closest to them and then drop it off at a station nearest to the user's end location. Due to this kind of system, the regular bike sharing systems fail to address the commuter's *last mile problem* in which commuters face the problem of being stuck in a place in between their destination location and the bike station to justify the effort of picking up and dropping off the bike. However, with their ability to be station-less, the station-less bike sharing systems have been successful in addressing the commuter's *last-mile problem*. With station-less bike sharing systems, commuters don't have to face the *last mile problem* of having to pick up the bike from a station and then park the bike back in a station at the end of the trip which may or may not be close to their original starting or destination location. Station-less bike sharing systems offer commuters the flexibility of picking up a bike from any location and then at the end of the trip just park the bike at a location most convenient to them. The whole system is managed using an app on the phone. Users can check for any available bikes near them using the pre-installed station-less bike app on their phone, the app displays bikes nearest to them. The user then picks a bike from the available ones for his/her trip. After the completion of the trip the user can park the bike at any point and then lock it, which signifies the end of the trip. The app records details of the trip like check-in/out time and the distance traveled based on which a fee will be charged.

Along with all of these advantages, due to the distinct and unique nature of station-less bike sharing systems where the user gets to pick up and drop off a bike at any location, they are facing new challenges which need to be addressed and solved. Three of the primary challenges are identifying regions within the city suffering from demand shortage of bikes, supply shortage of bikes and regions with parking problems. Since a user can pick up and drop off a station-less bike at any arbitrary location, the task of identifying regions suffering from these problems has become even more of a challenge. As a result busy streets of cities like Beijing are being flooded by thousands of bikes parked everywhere as shown in 4.1. This in turn is adding to the already existing parking and traffic congestion problems.

The main contributions of the developed framework are:

- We develop a multidimensional model to address some of the major problems being faced by the station-less bike sharing systems.

- Our developed framework can successfully identify problem areas within the city for different periods of time. Our model also incorporates a clustering model in addition to multiple dimensions.

Figure 4.1. Streets of Beijing overflowing with bikes

- In-order to deal with this kind of high amount of data as well as data sparsity we implement tensor factorization.
- We evaluate our model over Mobike datasets with more than 3.2 million trips in the city of Beijing. The experimental results verify the effectiveness of the developed model compared with baseline models.

4.2. DATA DESCRIPTION

In this section, we provide details about the Mobike and POI data sets that we have used in developing our multidimensional tensor model based on Tensor Factorization. Table 4.1 shows the statistics of our real-world data sets.

The Mobike trip dataset was released by Mobike in the Mobike Big Data Challenge 2017. The dataset is from the time period of May-10 to May-24 of 2017. It contains details of 3,214,096 trips along with 7 attributes associated with them. For each trip the associated attributes include details like the userid, orderid, trip start_time, trip start_location, trip end_location, etc. The initial data analysis revealed that the 84.29% of the entire dataset was made up by loyal users of Mobike who had regular routes and fixed patterns of bike usage. Out of the remaining 15.71% of the data which was made up by new/fickle users, 2.92% of the users had similar bike usage patterns and routes as the loyal users, and the remaining 12.83% of the data was generated by new/fickle customers who did not have the same bike usage patterns and routes. With these results we were able to conclude that 87.22% of the data had similar pattern. During the data analysis we considered users who rented bikes more than 7 times during the entire 14 days time period as loyal users and the users who rented bikes less than 7 times were categorized as new/fickle customers. The statistics from the analysis shown in Figure 4.2 drove us to the conclusion that the bike usage data had a consistent pattern and can be used to our advantage while identifying demand shortage regions, supply shortage regions and regions with parking problem.

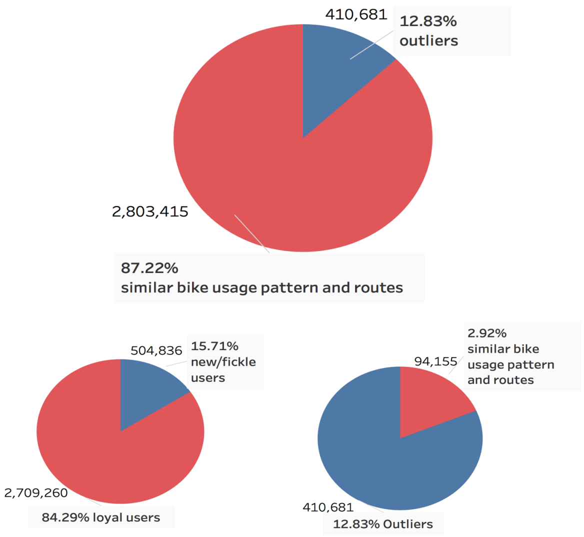


Figure 4.2. Data content identification

The POI data set for the city of Beijing was obtained from www.dianping.com, which is a commercial review and recommendation website. It consists of 328,668 POI's divided into 20 different categories like Hospitals, Malls, Restaurant, theaters, etc. We use the POI dataset to cross reference it with Mobike data and identify high activity regions within the city.

Table 4.1. Statistics of the datasets.

Data Sources	Attributes	Statistics
Mobike Trips	Number of trip records	3,214,096
	Number of users	349,693
	Trip start time	
	Trip start location	
	Trip end location	
	Bike id	
	Time period of records	05/10/2017 - 05/24/2017
	Number of POIs	328,668
POIs	Number of POI categories	20

4.3. PROBLEM DEFINITION AND FRAMEWORK

In this section we provide some key definitions and also give a brief overview of the developed framework.

4.3.1. Problem Definition. We initially state the below definitions to help break down the problem for better understandability. We then go on to give a more formal definition of the problem as well.

Definition 1: *(Trip):* A bike trip is defined as $Trip = \{T_{id}, T_{Sloc}, T_{Eloc} T_t\}$, where T_{id} denotes the trip's unique order id, T_{Sloc} consists of the latitude and longitude point of the trip's starting location, similarly T_{Eloc} consists of latitude and longitude point of trip's ending location, and T_t denotes the trip' start time.

Definition 2: *(Bike Supply Shortage Region):* A region r is defined as a bike supply shortage region S_r at time t if the number of CheckIn bikes is smaller than the number of CheckOut bikes.

Definition 3: *(Bike Demand Shortage Region):* A region r is defined as a bike demand shortage region D_r at time t if the number of CheckIn bikes is greater than the number of CheckOut bikes.

Definition 4: *(Bike check-in Tensors):* The three dimensional bike check-in tensor can be denoted as $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$. The check-in tensor contains data about the number of bikes checked in at a particular region for each day and hour.

Definition 5: *(Bike check-out Tensors):* The three dimensional bike checkOut tensor can be denoted as $\mathbf{Y} \in \mathbb{R}^{I \times J \times K}$. The check-out tensor contains data about the number of bikes checked out from a particular region for each day and hour.

Definition 6: *(Bike Parking Problem Region):* A region r is defined as a bike parking problem region P_r at time t if the total number of bikes present in region r at time t is greater than the predefined threshold P_{th}.

Problem Definition. Given a dataset consisting of bike trips along with their origin location, destination location, trip start time, order id, user id, and POIs of a city, our objective is to identify regions of the city with bike demand/supply shortage and parking problems.

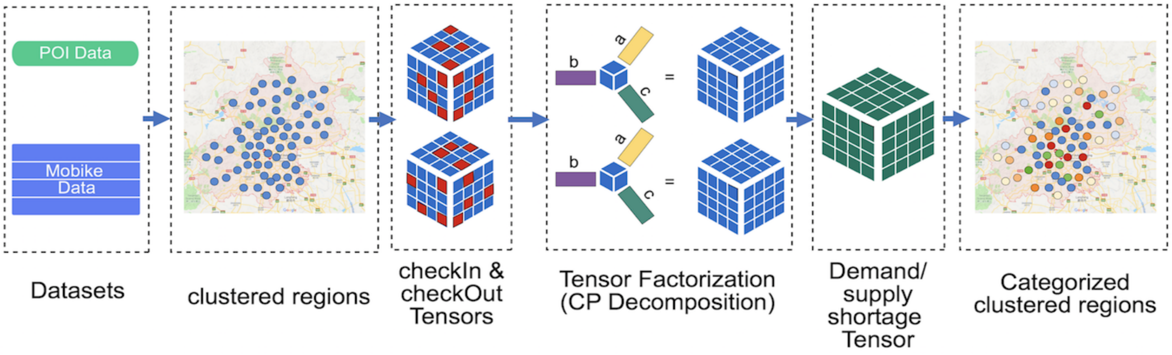

POI Data
Mobike Data
Datasets
clustered regions
checkIn & checkOut Tensors
b
a
c
=
Tensor Factorization (CP Decomposition)
Demand/ supply shortage Tensor
Categorized clustered regions

Figure 4.3. Framework of the developed model

4.3.2. Brief Overview of the Framework. Figure 4.3 shows the framework of our developed model. It consists of 4 major steps: preliminary data analysis and clustering model, tensor construction and factorization, identification of problem areas and optimization.

During the preliminary data analysis stage we leverage the POI data for the city of Beijing along with the bike check-in/out data from the Mobike dataset and cross-reference both of them to identify regions of the city with high activity. Then we construct 2 three dimensional tensors, one for bike check-in data and the other for bike check-out data. Here, regions, days and hours make up the 3 dimensions of the tensors. This enables us to capture the check-in/out data of every region for each hour of each day. In the identification of problem areas step we use the two constructed tensors to identify total number of bikes present in each region at specific times. We then use the resulting tensor to identify problem areas. Since we are dealing with multidimensional data the parameters required to be stored in a tensor can increase exponentially; additionally, to deal with the data sparsity problem we develop using the tensor factorization method.

4.4. CONSTRUCTING MULTIDIMENSIONAL MODEL

We use context information to construct our multidimensional model.

4.4.1. Preliminary Data Analysis and Clustering. The flow of data analysis and clustering is shown in Figure 4.4. We use the POI dataset for the city of Beijing to first identify regions with highest number of POI's in the city. The regions with higher number of POIs are considered as high human activity regions. We then also analyze the bike check-in and check-out data to identify regions which have high bike activity. We then cross-reference the high human activity regions with high bike activity regions which enables us to identify regions with highest activity in the city of Beijing in terms of POI's and bike activity. After identifying the highest activity regions we implement our clustering model on them to group the locations within the regions into clusters, with each cluster having a radius of 500 mt radius. Each cluster contains n number of both POI locations and bike check-in, check out locations.

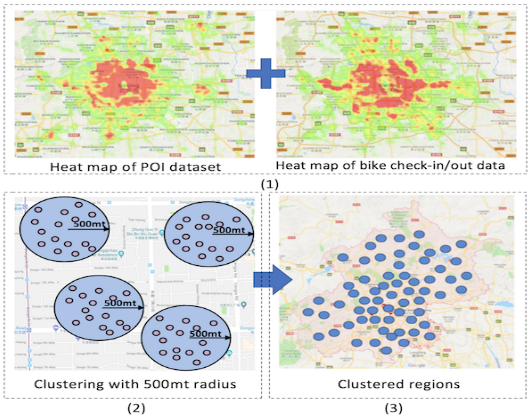


Figure 4.4. Clustering model

4.4.2. Tensor Construction. Based on the clustered model we take into account clustered regions which can also be considered as virtual stations to construct 2 tensors, one for bike check-in data and the other for bike check-out data as shown in Figure 4.5.

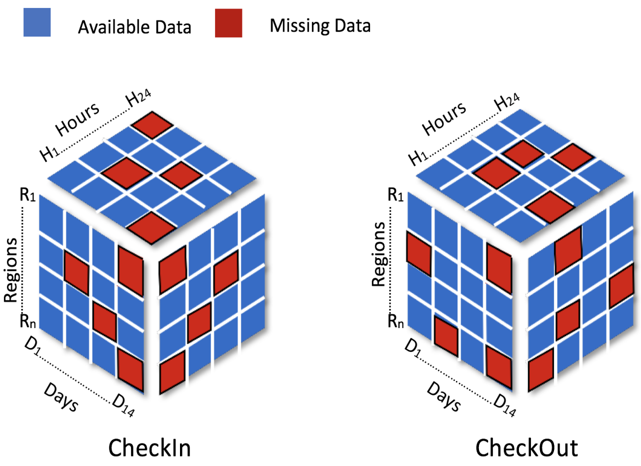


Figure 4.5. Check-in and check-out tensors

For both check-in and check-out tensors we represent each clustered region $i_1, i_2....i_n$ as the first dimension. We represent days $j_1, j_2....j_n$ and hours in a day $k_1, k_2....k_n$ as second and third dimensions. Check-in tensor $\mathbf{X}$ is represented as $\mathbf{X} \in \mathbb{R}^{I\times J\times K}$ and check-out tensor $\mathbf{Y}$ is represented as $\mathbf{Y} \in \mathbb{R}^{I\times J\times K}$.

4.4.3. Tensor Factorization. In-order to account for the missing data and to compress the data we implement tensor factorization using CP decomposition method[64]. We use the SiLRTC algorithm[65] in-order to detect missing values in a tensor. With this method we are not only able to retrieve the missing data but also will be able to store the data in its compressed form which can be retrieved later on for any future operations. The CP decomposition method factorizes a tensor into sum of a finite number of rank-one tensors. For example, given a third-order tensor $\mathbf{M} \in \mathbb{R}^{I\times J\times K}$ we can write it as

$$\mathbf{M} \approx \sum_{r=1}^{R} a_r \circ b_r \circ c_r \tag{4.1}$$

where R is a positive integer and $a_r \in \mathbb{R}^I$, $b_r \in \mathbb{R}^J$, and $c_r \in \mathbb{R}^K$. This can also be written elementwise as

$$m_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{4.2}$$

$$for\ i = 1, \ldots, I, j = 1, \ldots, J, k = 1, \ldots, K.$$

4.4.4. Fusing Tensors for Problem Area Identification. By fusing the two completed tensors we can identify demand/supply shortage regions and regions suffering from parking problems. By fixing the two dimensions of the tensor and performing matrix subtraction we are able to identify regions with demand/supply shortage problems. Now, in-order to identify regions suffering from parking problems we analyze the data based on the trip start time. By doing so, we were able to identify a pattern between bike usage and particular hours in a day. Figure 4.6 shows the usage pattern.

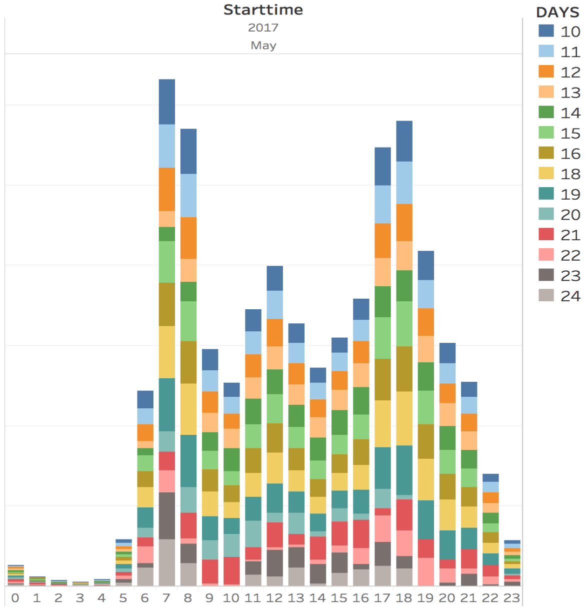


Figure 4.6. Bike usage pattern

We observed that the bike usage is high during 7AM and 8AM in the morning, and during 5PM, 6PM and 7PM in the evening. This is probably because of the number of people going to work and coming back from work after a night shift in the morning and people returning back home after work or people going to watch a movie or eat at a restaurant after getting off from work in the evening. This can be true not only for people who are using bikes to commute but also for people who are using taxis, buses and other means of transportation. We also observe a slight rise in bike usage from 11AM to 1PM. This might be due to people going out for lunch during break and people coming in for their afternoon shift at work. It is safe to assume that during these hours the probability of a region suffering from traffic congestion as well as a parking problem is at its peak.

We detect regions with parking problems by initially setting a parking threshold R_{pt}. We set a parking threshold while taking into consideration the fact that we're only analyzing the dataset of Mobike and that there are multiple companies like Mobike offering similar services whose bikes may also be present in the region being analyzed. We calculate parking threshold for individual regions depending on the growth rate of bikes parked in those regions. Given a region r and an initial parking threshold Pi_{th} with number of bikes parked in region r at hour 0 denoted as $r_i h_0$ and number of bikes parked in region r at hour 23 denoted as $r_i h_{23}$; we can then determine parking thresholds of individual regions:

$$\begin{cases} R_{1pt} = Pi_{th} + \{[(r_1 h_0 / r_1 h_{23})^{1/24} - 1] \times Pi_{th}\} \\ R_{2pt} = Pi_{th} + \{[(r_2 h_0 / r_2 h_{23})^{1/24} - 1] \times Pi_{th}\} \\ R_{3pt} = Pi_{th} + \{[(r_3 h_0 / r_3 h_{23})^{1/24} - 1] \times Pi_{th}\} \\ R_{4pt} = Pi_{th} + \{[(r_4 h_0 / r_4 h_{23})^{1/24} - 1] \times Pi_{th}\} \\ . \\ . \\ . \\ R_{n-1pt} = Pi_{th} + \{[(r_{n-1} h_0 / r_{n-1} h_{23})^{1/24} - 1] \times Pi_{th}\} \\ R_{npt} = Pi_{th} + \{[(r_n h_0 / r_n h_{23})^{1/24} - 1] \times Pi_{th}\} \end{cases} \tag{4.3}$$

The generic form of the equation can be written as,

$$R_{ipt} = Pi_{th} + \{[(r_i h_0 / r_i h_{23})^{1/24} - 1] \times Pi_{th}\} \tag{4.4}$$

The parking threshold R_{ipt} value differs for different regions. The region r is considered as a region suffering from parking problem P_r at time t if the total number of bikes present in that region is greater than the predefined threshold. Algorithm 1 gives the details of the procedure. Based on this we are also able to determine that parking problem of a region is independent of supply shortage and demand shortage factor of a region. It is solely dependent on the predefined threshold and individual hours of a day.

Algorithm 1 Supply, Demand and Parking problem regions detection

Input: CheckIn tensor **X**, checkOut tensor **Y** and set of parking thresholds p for different categories

Output: Identification of demand shortage regions D_r, supply shortage regions S_r and parking problem regions P_r

Initialize tensors **X** and **Y** after fixing any two dimensions d_0, d_1 provided both fixed dimensions d_0 and d_1 are same for the two tensors.

```
if $d_1, d_2$ of $\mathbf{X}$ == $d_1, d_2$ of $\mathbf{Y}$ then
    $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$
    if $\mathbf{Z}_{i,j,k} > 0$ then
        $Z_{i,j,k} = D_r$
    end
    else if $\mathbf{Z}_{i,j,k} < 0$ then
        $Z_{i,j,k} = S_r$
    end
    for $p_1, p_2, p_3 ... p_n$ do
        for $f$ in $[1, k]$ do
            if $\mathbf{r}(Z_{i,j,k}) > p$ then
                $r(Z_{i,j,k}) = p_r$
            end
        end
    end
end
```

4.5. MODEL EVALUATION

We evaluate our model for two criteria: (i) We implement tensor factorization on different ranks to test which rank gives the optimal accuracy while keeping the number of parameters to be stored at an acceptable limit when the tensor is decompressed. (ii) We test our model against different methods to examine the accuracy of the predicted missing values recovered by implementing tensor factorization. Based on this we determine its effectiveness.

Root Mean Square Error (RMSE)[66] measures how much error there is between two data sets. In other words, it compares a predicted value and an observed or known value. The formula for calculating RMSE is as below.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(y_j - \hat{y}_j\right)^2}$$

It quantifies how different a set of values are. The smaller an RMSE value, the closer predicted and observed values are.

In-order to find the optimal rank we factorize our tensors on different ranks and then check the RMSE error between the original tensor and the recovered tensor. In the experiment we implement tensor factorization as well as matrix factorization [67] on the data set while keeping the rank same for both the methods.

Table 4.2. RMSE values for different ranks

Rank	Matrix Factorization	Matrix Factorization	Tensor Factorization	
	RMSE	Parameters	RMSE	Parameters
1	5.04315	1286	3.16346	988
2	3.54652	2572	2.75363	1976
3	3.47987	3858	2.50851	2964
4	2.61652	5144	2.22143	3952
5	2.57374	6430	2.12987	4940

The experimental results shown in Table 4.2 lead us to conclude that tensor factorization not only provides better accuracy while reconstructing the original tensor from the factorized tensor, but can also keep the number of parameters to be stored at minimal when compared with matrix factorization.

For the second criteria we evaluate our model against other models to determine its effectiveness. We test our model against Matrix Factorization and Linear Regression for different percentage of missing data.

Matrix Factorization is one of the most widely used methods to predict missing values and also to compress the data. Models based on Matrix Factorization(MF) have received greater exposure, mainly as an unsupervised learning method for latent variable decomposition and dimensionality reduction [68][45]. It is most similar to the tensor factorization method. In MF, a matrix V is factorized into two matrices W and H and the missing values are approximated numerically.

Linear Regression is also one of the most widely used methods to predict missing values. Its broad appeal and usefulness results from a conceptually logical process of using an equation to express the relationship between a variable of interest and a set of related predictor variables. It uses the relationship between scalar dependent variables and one or more explanatory variables to predict missing values[69].

The main aim of this evaluation is to check which model can better predict the missing data. We measure accuracy using RMSE method. In the first case where there is only 10% of the data is missing, the RMSE value is considerably low, but as we increase the percentages of deleted data the RMSE value also increases. In the final test case where we test against 50% of the missing data, we can see that RMSE value increases considerably more. In all the test cases, we observe that RMSE value of tensor factorization manges to be lower when compared with values of other methods and based on this we conclude that our model performs better than the other two methods. Figure 4.7 shows the result of our experiment.

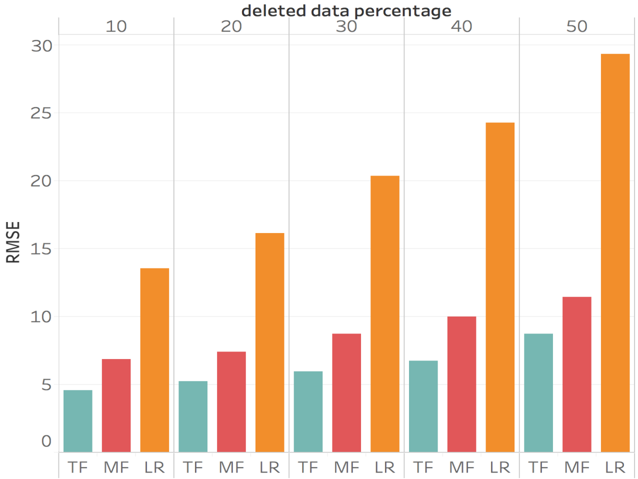


Figure 4.7. RMSE values of TF, MF and LR for different missing data percentage

4.6. CONCLUSION

In Section 4, we develop a framework which combines a cluster model and a multidimensional tensor based model to address some of the major problems faced by station-less bike sharing systems. We use our model to predict and detect areas with demand/supply shortage of bikes and areas with parking problems. In-order to achieve this, we combine Mobike dataset for the city of Beijing and the POI dataset to build our model. We first use the POI dataset along with bike check-in and check-out data to identify regions with high activity. We then use clustering model to divide these high activity regions into virtual stations. Then we construct our multidimensional tensor based model. We start construction of our model by building two separate tensors, one for check-in data and the other for check-out data of bikes, then we implement tensor factorization to predict any missing values within the data and to compress data. Final steps consist of fusing the two tensors which enables us to successfully identify regions with demand/supply shortage of bikes and parking problems. The unique perspective of our model is due to the combination of clustering and the multidimensional tensor model. Experimental results show the effectiveness of our model.

5. DISTRIBUTED REPRESENTATION LEARNING FRAMEWORK FOR LEARNING HUMAN MOBILITY PATTERNS

To address the problem of learning representations of spatiotemporal heterogeneous networks, in this section, two types of collective representation learning models for learning distributed representations of Spatiotemporal Heterogeneous Network Embedding (SHNE) are presented. (i) Multilingual SHNE (M-SHNE): the developed model leverages the use of random walks along with a multilingual word embedding technique used in natural language processing (NLP) to collectively learn the spatiotemporal proximity measures between vertices in Spatiotemporal Heterogeneous Networks (SHNs) and preserve it in a low dimensional vector space. (ii) Meta-path Constrained Random walk SHNE (MCR-SHNE): this combines the advantage of meta-path counting algorithm, path constrained random walks, and a word embedding technique to generate lower dimensional embeddings that preserve the spatiotemporal proximity measures in Spatiotemporal Heterogeneous Networks (SHNs). Experimental results demonstrate the effectiveness of the two developed models over state-of-the-art algorithms on real-world datasets.

5.1. BACKGROUND AND OVERVIEW

Lately, new network embedding models which consider the interconnected, multi-typed properties of heterogeneous networks have been developed. Some of the existing network embedding models [12, 13] develop structural analysis approaches by leveraging the rich semantic meaning of structural types of objects and links in the networks. Heterogeneous networks embody a vast number of interrelated facts, and they can facilitate the discovery of interesting knowledge [70]. Figure 5.1 illustrates a SHN of a region within the Beijing city.

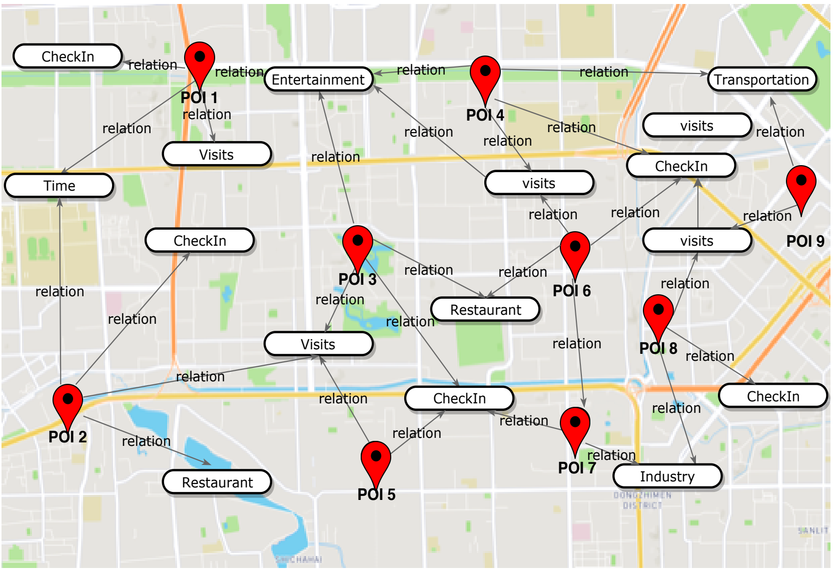


Figure 5.1. Spatiotemporal Heterogeneous Network

Here, we consider POIs (points of interest) as vertices and edges between vertices represent multiple spatial and temporal relationships between POIs of different class types (e.g., POI categories, distance, region, day and time of visit, Check-In, Check-Out, etc.). The proximity among vertices in a heterogeneous network is not just a measure of closeness or distance, but it is dependent on semantics as well. For example, the SHN in Figure 5.1 represents POIs in a city and multiple relationships connecting them. POI_1 is connected to both POI_2 and POI_3, but the proximity between them differs. From Figure 5.1 we can observe that POI_1 and POI_2 are connected by day of visits whereas POI_1 and POI_3 have much stronger connection as not only are they connected by day of visits, they also belong to the same POI category and are geographically situated close to each other.

In this section, we develop two distributed representation learning-based methods for collectively embedding the spatial and temporal properties of SHNs. (i) Multilingual SHNE (M-SHNE): this method treats a SHN as a set of individual homogeneous networks where each homogeneous network contains information about a type of relationship (distance, time of visits, category, etc.) that exists between vertices. Random walks are then used to capture the latent structure of these individual homogeneous networks and generate a corpus. The generated corpora of all the homogeneous networks in the set are then treated as a set of multilingual corpora where each corpus is a translation of the same network but in a different language. The model then leverages a multilingual word embedding technique used in NLP [71] to learn the embeddings of the SHN as a whole. (ii) A second method called Meta-path Constrained Random walk SHNE (MCR-SHNE) is developed. This method captures the relationship between vertices in an SHN by leveraging the use of path constrained random walks (PCRW) [16] and a meta-path counting algorithm [14, 53]. A refined corpus which encompasses the different relationships between vertices in a SHN is then generated based on the path constrained random walks and meta-path counting algorithm. It then learns the embedding of the SHN by using a language embedding technique on the generated corpus. Experiments are conducted on multiple real-world SHN datasets to compare our two developed methods with state-of-the-art network embedding methods (i.e., LINE [13], Deep- Walk [9], GraRep [12] and node2vec [10]) on classification and clustering tasks. Our experimental results show that the two developed methods outperform all alternative approaches in most of the qualitative measures used.

Overview. The remainder of Section 5 is organized as follows. In Section 5.2, we provide the problem definition for SHN embedding. Section 5.3 provides details about the frameworks of our two developed models. In Section 5.4, we report the experimental results of our two developed models for SHN embedding. Section 5.5 provides a brief review of related works. Finally, we conclude in Section 5.6.

5.2. PROBLEM DEFINITION

We initially provide the below definitions to help break down the problem for better understandability. We then give a formal definition of the problem.

Definition 1: *(Homogeneous Network (HN)):* A homogeneous network is represented as $G = (V, E)$, where $V = \{v_i\}_{i=1,\ldots,N}$ consists of a set of vertices, $e_{i,j} = \left(v_i, v_j\right) \in E$ is an edge that indicates the relationship between two vertices.

Definition 2: *(Network Embedding (NE)):*Network embedding aims to capture the latent structure of a network in a low dimensional representation $x_i \in \mathbb{R}^d$ for each vertex $v_i \in V$, where d is the dimension of the embedding space.

Definition 3: *(Spatiotemporal Heterogeneous Network (SHN)):* A SHN is a directed graph $G = (V, E, \phi, \psi)$ with an object type mapping function $\phi : V \longrightarrow \mathcal{L}$ and a link type mapping function $\psi : E \longrightarrow \mathcal{R}$, where each object $v \in V$ belongs to an object type $\phi(v) \in \mathcal{L}$ and each link $e \in E$ belongs to a link type $\psi(e) \in \mathcal{R}$. Here, POIs constitute vertices in the network and edges represent multi-class spatiotemporal relationships.

Definition 4: *(Meta-paths):* In a SHN G, two vertices v_0, v_1 may be connected via multiple edges. We call the connecting edges meta-paths. Conceptually, each of these meta-paths represents a specific direct or composite relationship between them. In Figure 5.1, $POIs$ POI_3 and POI_4 are connected via multiple paths. For example, $POI_3 \longrightarrow c \longrightarrow POI_4$ path represents a relationship that the POI_1 and POI_3 belong to the same POI category. $POI_3 \longrightarrow c \longrightarrow POI_6 \longrightarrow v \longrightarrow POI_4$ indicates that POI_3 is connected to POI_4 via POI_6 where POI_3 and POI_6 belong to the same POI category and POI_6 and POI_4 are connected via day of visit.

Definition 5: *(Multilingual Embedding):* Multilingual word embeddings represent words of multiple languages embedded in the same vector space and allow the transfer of knowledge from one language to the other without machine translation.

Problem Definition. Given a spatiotemporal heterogeneous network (SHN) $G = (V, E, \phi, \psi)$. We formulate a function $f : V \longrightarrow \mathbb{R}^d$ that projects each vertex $v \in V$ to a vector in d dimensional space $\mathbb{R}^d$, such that the multiple different classes of spatial and temporal relationships that exist between vertices of a SHN are also preserved.

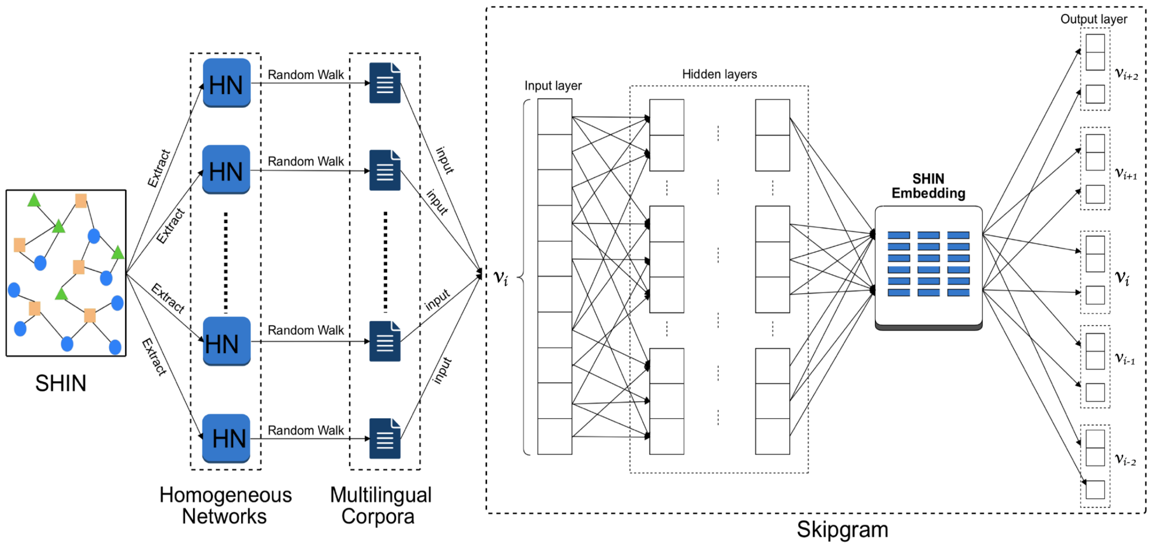


Figure 5.2. Framework of Multilingual SHNE model

5.3. DEVELOPED MODELS

In this section, we describe our two developed models. In Section 5.3.1, we introduce our first method M-SHNE. Then, we introduce our second model MCR-SHNE in Section 5.3.2.

5.3.1. Multilingual SHN Embedding (M-SHNE) . The framework of the developed Multilingual SHN Embedding (M-SHNE) method is as shown in Figure 5.2. The model accepts a SHN as input and separates it into a series of homogeneous networks, where each homogeneous network represents a different relationship between vertices. It then performs random walks on the extracted homogeneous networks to capture the underlying latent structure. Here each random walk pass is treated as a sentence, and in this way, a corpus comprised of multiple passes of random walks is generated for each homogeneous network. The model treats these generated set of corpora as multilingual, i.e., where each corpus represents the same space and the same set of POIs as vertices but encompasses different relationships between vertices. Multilingual word embedding technique is then used to learn SHN embeddings from the multilingual corpora.

5.3.1.1. Homogeneous network extraction. The model analyzes the SHN to identify different individual relationships that exist between the vertices. For each link $e \in E$ that belongs to a link type $\psi(e) \in \mathcal{R}$, it extracts and creates a series of homogeneous networks which is the same as the number of identified relationships within the SHN. Each extracted homogeneous network is of the same space and contains the same number of vertices, but they only encompass a single-class of spatial or temporal relationship.

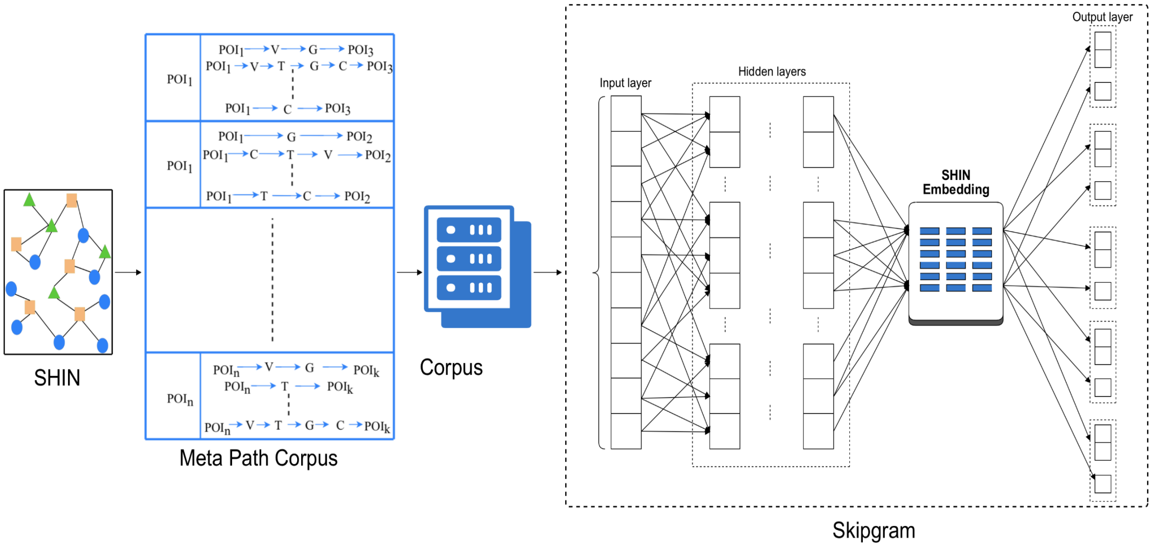

SHIN
POI$_1$
POI$_1$ → V → G → POI$_3$
POI$_1$ → V → T → G → C → POI$_3$
POI$_1$ → C → POI$_3$
POI$_1$
POI$_1$ → G → POI$_2$
POI$_1$ → C → T → V → POI$_2$
POI$_1$ → T → C → POI$_2$
POI$_n$
POI$_n$ → V → G → POI$_k$
POI$_n$ → T → POI$_k$
POI$_n$ → V → T → G → C → POI$_k$
Meta Path Corpus
Corpus
Input layer
Hidden layers
SHIN Embedding
Output layer
Skipgram

Figure 5.3. Framework of MCR SHN Embedding

5.3.1.2. Multilingual corpora generation. The model performs random walks on the extracted homogeneous networks to capture their latent structures. Here, the model considers each random walk pass as a sentence and builds a corpus based on the sentences generated from multiple random walks on each vertex of the network. The set of corpora generated from these homogeneous networks is considered as a multilingual corpora as each corpus in the set represents the same network space and vertexes. The only difference is the way random walks capture the spatial and temporal contextual relationships between the vertices.

5.3.1.3. SHN embedding. We use skip-gram for Multilingual SHN Embedding. It is a language model that maximizes the co-occurrence probability among the words that appear within a window, w, in a sentence [11]. Specifically, we use an objective function shown in Equation 5.1 to map the lexical vector $\mathcal{T}$ of multilingual corpora to its semantic space E.

$$E\left(\mathcal{T}\right) = \frac{\sum_{v_o \in v_{lex}(\mathcal{T})} \left(\frac{1}{(v_o, v_{lex})(\mathcal{T})} E\left(v_o\right)\right)}{\sum_{v_o \in v_{lex}(\mathcal{T})} \left(\frac{1}{(v_o, v_{lex})(\mathcal{T})}\right)} \tag{5.1}$$

where, E $\left(v_o\right)$ is the embedding-based representation of the vertex v_o in E, and $\left(v_o, v_{lex}\right)$ is the vertex dimension corresponding to the v_o in the lexical vector $v_{lex}\left(\mathcal{T}\right)$.

The objective function helps us preserve the spatiotemporal heterogeneities influence on data semantics and meaning in the SHN embedding by mapping lexical vectors and extracting semantic information. Algorithm 2 demonstrates the process of SHN embedding.

Algorithm 2 Multilingual Heterogeneous Network embedding

```
Input: graph $G = (V, E, \phi\psi)$
       window size $\omega$
       embedding size $d$
       walks per vertex $\gamma$
       walk length $t$
Output: matrix of vertex representation $\Phi \in \mathbb{R}^{|V| \times d}$
Initialization: Extract $g(V, E)$ from $G = (V, E, \phi, \psi)$ for each $v_i \in G$ do
    for $i = 0$ to $\gamma$ do
        select $g_i \in G$ with prob $\frac{1}{|g|}$
        $\mathcal{W}_{v_i}$=RandomWalk$(g, v_i, t)$
    end
    $\mathcal{W}_g \leftarrow W_{v_i} + 1$
    for each $W_{v_i} \in \mathcal{W}_g$ do
        for each $v_j \in W_{v_i}$ do
            for each $u_k \in W_{v_i}\ [j - w : j + w]$ do
                $J\left(\Phi\right) = -logPr\left(u_k | \Phi\left(v_j\right)\right)$
                $\Phi = \Phi - \alpha * \frac{\partial j}{\partial \Phi}$
            end
        end
    end
end
```

The M-SHNE first extracts homogeneous networks $g\ (V, E)$ from SHN $G = (V, E, \phi, \psi)$. For each $v_i \in G$ we perform γ random walks on the selected homogeneous network $g_i \in G$, where $G = \{g_1, g_2, ..., g_n\}$ and generate a corpus $\mathcal{W}_g$ of the network. The skip-gram then iterates over each corpus $\mathcal{W}_g$ and all possible collocations in the random walk that appear within the window w of the corpora. It maps each vertex v_i to its current representation vector $\Phi(v_i) \in \mathbb{R}^d$. It updates the generated SHN embedding over every iteration by mapping lexical vectors to their semantic space and preserves complex proximity measures that exist between vertices in a heterogeneous network.

5.3.2. MCR SHN Embedding (MCR-SHNE). The framework of the developed Meta-path Constrained Random walk SHN Embedding method (MCR-SHNE) is as shown in Figure 5.3. The model takes a SHN as input. Path constrained random walks are performed on the SHN to identify and capture several meta-path instances that exist between vertices. A path counting algorithm then uses identified meta-path instances to calculate the proximity measures between vertices. Based on this, a refined meta-path based corpus encompassing multiple meta-path instances of multi-class spatial and temporal relationships between vertices is then formed. Word embedding technique is then used on the meta-path based corpus to learn embeddings of SHN.

5.3.2.1. Meta-path corpus generation. Here we adopt a slightly modified version of random walks called path constrained random walks (PCRW) for corpus generation. In addition to preserving proximity measures as random walks, PCRW also encapsulates information about the multiple meta-path instances that exist between vertices. Table 5.1 shows a few meta-path instances that exist between POIs, according to the SHN in Figure 5.1. We then generate a refined meta-path based corpus based on the proximity scores calculated by PCRW.

The meta-path counting algorithm calculates the proximity between two vertices v_s, v_t as:

$$s\,(v_s, v_t) = \sum s\,(v_s, v_t | \mathcal{P}) \tag{5.2}$$

where, $s(v_s, v_t|\mathcal{P})$ is the proximity score between vertices v_s to v_t.

It is calculated based on the probability that a PCRW path restricted on a meta-path would follow the instance $p_{v_s \to v_t}$. We can calculate the probability score between a pair of vertices $v_s, v_t \in V$ based on the meta-path instances generated from PCRWs as:

$$s(v_s, v_t|\mathcal{P}) = \sum_{p_{v_s \to v_t} \in \mathcal{P}} s\left(v_s, v_t|p_{v_s \to v_t}\right) \tag{5.3}$$

where $s\left(v_s, v_t|p_{v_s \to v_t}\right)$ is the proximity score w.r.t mata path instance $p_{v_s \to v_t}$ between (v_s, v_t).

Table 5.1. Meta-paths between POIs in SHN of Figure 5.1

Meta-Path	**Semantic Meanings**
$POI_1 \to t \to POI_2$	POI_1 and POI_2 open at same time
$POI_3 \to e \to v \to POI_4$	POI_3 and POI_4 belong to same category and are visited at same time
$POI_4 \to c \to v \to POI_8$	POI_4 and POI_8 on average have same number of people checking in and are visited at same time
$POI_5 \to c \to v \to e \to POI_4$	POI_5 and POI_8 belong to same category, have same number of people checking in on average and are visited at same time

5.3.2.2. SHN embedding. Algorithm 3 shows the steps for generating MCR based SHN embedding. Here, we use short truncated random walks of length t and meta-path length l along with path constraint factor C to control the length of the meta-path. We do this because shorter meta-paths are more informative than longer ones; longer meta-paths tend to link more remote vertexes which might be less contextually related [14]. The goal here is to create a SHNE using meta-path based corpus in a way that given a SHN $G = (V, E)$, we develop an embedding that transforms each vertex $v \in V$ to a vector in $\mathbb{R}^d$, such that the proximities between any two vertices in the original heterogeneous networks are preserved in $\mathbb{R}^d$. The MCR-SHNE performs PCRW walks on each node $v_s \in G$ based on the walks per vertex γ , walk length t, meta-path length l, and path constraint C hyperparameters set on it and generates meta-path based corpus. The meta-path based corpus is then passed as input to skip-gram. The skip-gram then iterates over all possible collocations of PCRW that appear within the window w of the corpus $\mathcal{M}_v$. It maps each vertex v_j to its current representation vector $\Phi(v_i) \in \mathbb{R}^d$ and generates SHNE that preserves complex spatial and temporal proximity measures between vertices in an SHN.

Algorithm 3 MCR SHNE

```
Input: graph $G = (V, E, \phi, \psi)$
       window size $\omega$
       embedding size $d$
       walks per vertex $\gamma$
       path constraint $C$
       walk length $t$
       meta-path length $l$
Output: matrix of vertex representation $\Phi \in \mathbb{R}^{|V| \times d}$

for each $v_s \in G$ do
    for $i = 0$ to $\gamma$ do
        Rlength = 0
         Mlength = 0
         if $Rlength < t$ && $Mlength < l$ then
             $p_{v_s \rightarrow v_t}$ = select $v_t$ with prob $\frac{1}{|v_t \in C|}$
              $\mathcal{M}_v \leftarrow p_{v_s \rightarrow v_t} + 1$
              $Mlength \leftarrow Mlength + 1$
              $Rlength \leftarrow Rlength + 1$
        end
        for each $v_j \in \mathcal{M}_v$ do
            $\mathcal{W}_{v_i} = \mathcal{M}_v \rightarrow \mathcal{M}_v + 1$
            for each $u_k \in W_{v_i}$ $[j - w : j + w]$ do
                $J(\Phi) = -logPr\left(u_k|\Phi\left(v_j\right)\right)$
                $\Phi = \Phi - \alpha * \frac{\partial j}{\partial \Phi}$
            end
        end
    end
end
```

5.3.3. Negative Sampling. We also implement Negative sampling in the skip-gram part of both of our developed models. Negative sampling [11] is a loss function which rewards the estimate of the probability for vertices that co-occur with each other, while penalizing the estimate of the probability for random vertex pairs co-occurring with each other. Negative sampling has proven to be a very effective alternative to the computationally expensive softmax, where we need to sum the overall non-zero proximity scores $s(v_i, v_j)$ for a specific vertex v_i. Negative sampling has achieved state of the art results in many NLP tasks and plays a major role in substantially speeding up the learning process and helps generate better embeddings [72].

Equation 5 describes the loss function for each vertex context occurrence in a SHN corpus generated from our two developed models. Here, we consider $v \in V$ as the target vertex, $c \in C$ a context vertex and $c_n \in C$ as random negative sample (indexed by n) drawn from a noise distribution $p\left(c\right)$.

$$J_{sg}\left(v, c\right) = \log\left(\sigma\left(c^t v\right)\right) + \sum_{n=1}^{N} \mathbb{E}_{c_n \sim P(c)}\left[\log\left(\sigma\left(-c_n^t v\right)\right)\right] \tag{5.4}$$

The total loss is the summation of Equation 5.4 for all pairs of vertices v and c co-occurring in the corpus, as extracted using a context window of size w.

5.4. EXPERIMENTAL SECTION

We evaluate the performances of our developed models against baseline models on real-world SHNs of Beijing city.

5.4.1. Data Description. In this section, we provide details about the Mobike, POI, and Weibo & Jiepang datasets used for evaluation. Table 5.2 shows the statistics of the datasets mentioned above.

Table 5.2. Statistics of the datasets.

Data Sources	**Attributes**	**Statistics**
Mobike Trips	# trip records	3,214,096
	# users	349,693
	Trip start time	
	Trip start location	
	Trip end location	
	Bike id	
	Time period of records	05/10/2017 - 05/24/2017
POIs	Number of POIs	328,668
	# POI categories	20
Weibo & Jiepang	# Check-In	2,020,967
	# users	212,362
	POI Name	
	POI Location	
	POI address	

Mobike, the stationless bike-sharing company, released its Beijing city trip dataset in the 2017 Mobike Big Data Challenge. It contains details of 3,214,096 trips. It includes information about the number of users, trip start time, trip endtime, bike id, trip start location, and trip end location. The POI data set for the city of Beijing was acquired from www.dianping.com, which is a commercial review and recommendation website. It contains details about 328,668 POIs divided into 20 different categories like Hospitals, Malls, Restaurant, theaters, etc. The Weibo & Jeipang datasets together include 2,020,967 check-in entries of POIs in Beijing. It contains details like POI name, POI check-in time, POI address, and POI location.

5.4.2. SHN Construction. To evaluate the effectiveness of our two developed models on very large and complex SHNs, we construct two SHNs from Mobike, POI, and Weibo & Jeipang datasets. This section provides information about the steps followed to construct the two SHNs.

SHN1: For the construction of our SHN1, we utilize the Mobike and the POI datasets. We assigned 1,765,025 trip start and end locations with visits greater than 300 as vertices. We integrated the POI dataset into the Mobike dataset to assimilate the rich multi-class relationships that exist within the POI dataset into SHN1. To summarize, our

SHN1 consisted of 1,765,025 vertices and encapsulates multi-class spatial and temporal relationships like visits per day, visits per hour, geographic distance, 20 different POI categories, high activity time, and low activity time.

SHN2: To construct our SHN2, we use the Weibo & Jeipang dataset in combination with the POI dataset. We considered the 472,654 individual POIs present in the datasets as vertices. By combining the POI dataset, we also increased the number of multi-class spatial and temporal relationships that exist between vertices. To summarize, our SHN2 consisted of 472,654 vertices in total and encapsulates multi-class spatial and temporal relationships like visits per day, visits per hour, geographic distance, 20 different POI categories, number of individual user visits, high activity time, and low activity time.

5.4.3. Baseline Models and Experimental Settings. We consider four baselines to demonstrate the effectiveness and robustness of our two developed models.

- DeepWalk [9]: DeepWalk first transforms the network into node sequences by truncated random walk, and then uses it as input to the skip-gram model to learn representations.
- LINE [13]: LINE can preserve both first-order and second-order proximities for the undirected network through modeling node co-occurrence probability and node conditional probability.
- GraRep [12]: GraRep preserves node proximities by constructing different k-order transition matrices.
- Node2vec [10]: node2vec develops a biased random walk procedure to explore the neighborhood of a node, which can strike a balance between local properties and global properties of a network.

5.4.4. Parameter Setting. We evaluate our developed models against the baseline models on classification and clustering data mining tasks. We use OpenNE Python toolkit to implement the DeepWalk, LINE, GraRep, and node2vec models. For the DeepWalk and node2vec methods, we set the window size, walk length, and the number of walks on each vertex as 10, 40, and 10, respectively. For LINE we set the number of negative samples as 5, and the learning rate is set to be 0.025. For GraRep we set maximum matrix transition step K=6, and we also used L2 normalization.

For our two developed models, we set the number of negative samples per input sample *m* to 5, window size, walk length, and the number of walks on each vertex as 10, 40, and 10. For all models and SHNs, we set the embedding dimension to d = 80. We evaluate the performances of the models by calculating Macro-F1 and Micro-F1 scores.

5.4.5. Classification and Robustness Check. Classification in network analysis is an important task in many applications. We perform the task of multi-label classification using generated network embeddings of SHN1 and SHN2 from the developed models as well as the baseline models on a logistic regression classifier. For the task of classification, we imply network embeddings as vertex features and feed them into a logistic regression classifier model, and we use different POI categories as labels. To check the robustness of the models, we perform classification on the datasets with varying percentages of labeled nodes from 10% to 90%. Table 5.3 and Table 5.4 show classification accuracies with different training ratios on different datasets. The best results are bold-faced.

We observe that M-SHNE and MCR-SHNE methods perform better than baseline models in most of the cases. However, GraRep performs slightly better than our two developed models on two occasions where embeddings generated from SHN1 were used. A potential explanation for this could be the low number of labeled nodes available for our models to build corpus on and generate quality embeddings. On SHN2 our models outperform all the baseline models. To summarise, our developed models outperform baseline models and generate better network embeddings that preserve the multi-class relationships between vertices in a SHN. They also generate superior robust embeddings and consistently outperform all the other baselines for varying ratios of labeled nodes.

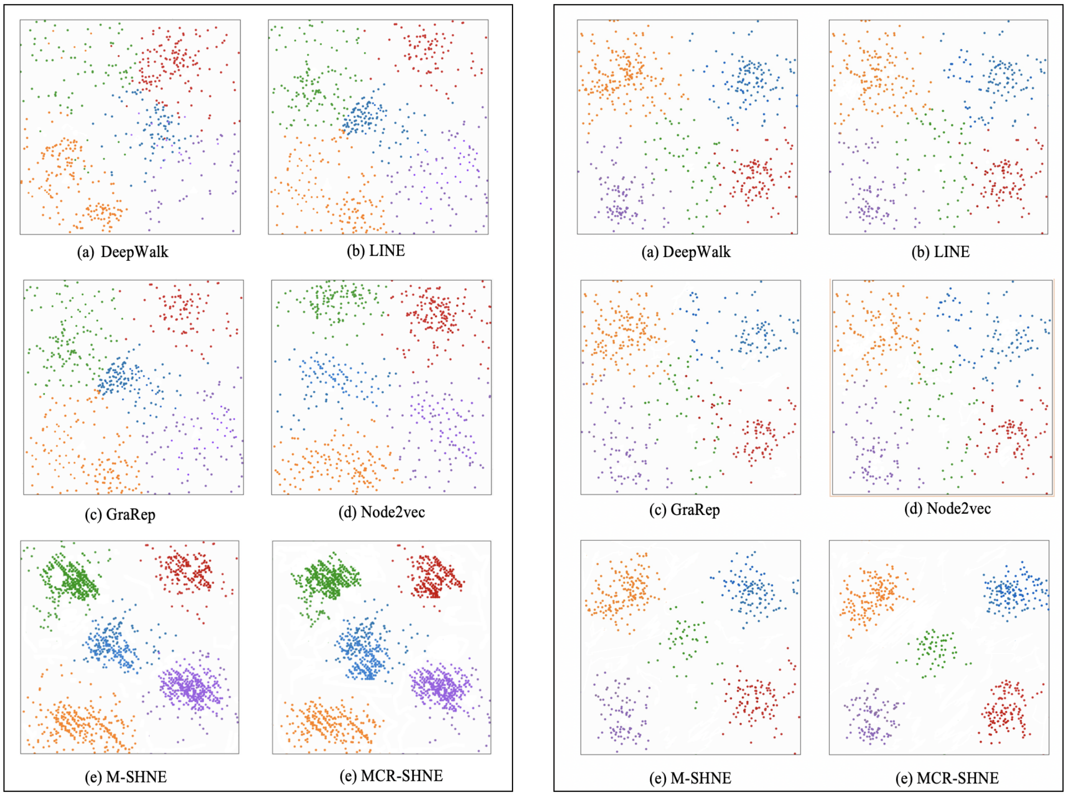
Table 5.3. Accuracy (%) of node classification on SHN1

	% Labeled Nodes	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Micro-F1%	DeepWalk	56.53	57.03	57.90	58.42	60.15	62.73	64.79	63.26	63.82
	LINE	53.84	54.51	55.14	55.87	56.22	56.94	57.27	58.43	58.94
	GraRep	**59.19**	**60.73**	60.57	59.65	60.37	61.76	62.64	63.52	64.16
	node2vec	55.71	55.03	56.90	58.39	60.14	60.84	61.75	62.93	62.21
	M-SHNE	57.34	58.93	59.67	61.84	62.24	63.88	63.27	64.04	65.82
	MCR-SHNE	58.63	59.85	**60.85**	**62.74**	**64.71**	**65.26**	**65.92**	**66.36**	**67.47**
Macro-F1%	DeepWalk	31.25	32.65	33.19	33.80	34.40	35.17	35.88	36.02	35.72
	LINE	28.30	29.85	30.90	30.33	31.47	31.65	32.45	33.73	33.57
	GraRep	33.44	34.62	35.02	35.89	36.32	36.78	35.25	36.45	37.57
	node2vec	28.68	29.81	30.54	32.12	32.94	33.87	34.03	34.79	35.78
	M-SHNE	33.14	34.67	35.33	**36.01**	36.87	37.32	36.57	**37.90**	38.34
	MCR-SHNE	**34.51**	**35.65**	**36.23**	35.31	**37.05**	**37.87**	**38.72**	37.61	**39.83**

Table 5.4. Accuracy (%) of node classification on SHN2

	% Labeled Nodes	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Micro-F1%	DeepWalk	46.26	47.92	48.53	49.12	51.28	52.40	53.23	55.06	56.49
	LINE	42.43	42.92	43.85	44.63	45.06	46.48	47.12	49.84	53.73
	GraRep	47.43	47.96	48.62	49.13	49.86	49.64	50.38	51.15	51.76
	node2vec	48.43	48.96	49.75	50.64	51.91	52.74	53.28	53.82	54.65
	M-SHNE	49.43	50.45	51.30	51.69	53.71	54.38	**56.18**	57.34	58.78
	MCR-SHNE	**50.29**	**52.12**	**52.55**	**53.39**	**54.77**	**55.02**	55.83	**57.95**	**59.02**
Macro-F1%	DeepWalk	29.96	30.18	30.75	31.93	32.38	33.13	33.63	34.70	35.18
	LINE	19.35	20.41	21.70	22.04	22.43	23.69	23.26	24.90	25.67
	GraRep	23.47	24.22	24.87	25.54	26.84	27.17	27.73	28.45	28.02
	node2vec	27.15	27.28	28.05	24.33	24.19	25.17	25.56	26.21	27.17
	M-SHNE	30.04	30.96	31.74	31.66	32.43	32.79	36.26	**37.83**	36.53
	MCR-SHNE	**31.24**	**32.11**	**34.02**	**34.89**	**35.08**	**36.34**	**37.74**	37.24	**38.56**

5.4.6. Clustering. Clustering is the data mining task of identifying natural groups in the data. Here, as we are dealing with SHNs of Beijing city, we aim to identify communities that share similar functionality and representations. For the sake of experimentation, we define a community as consisting of (i) a location (i.e., latitude and longitude) of a residential complex, and (ii) a neighborhood area (e.g., a circle with a radius of 1 km) consisting of POIs. Adhering to this definition, we identify around 1,023 communities in SHN1 and 712 communities in SHN2 made up of multiple POIs belonging to different



(a) Clustering on SHN1

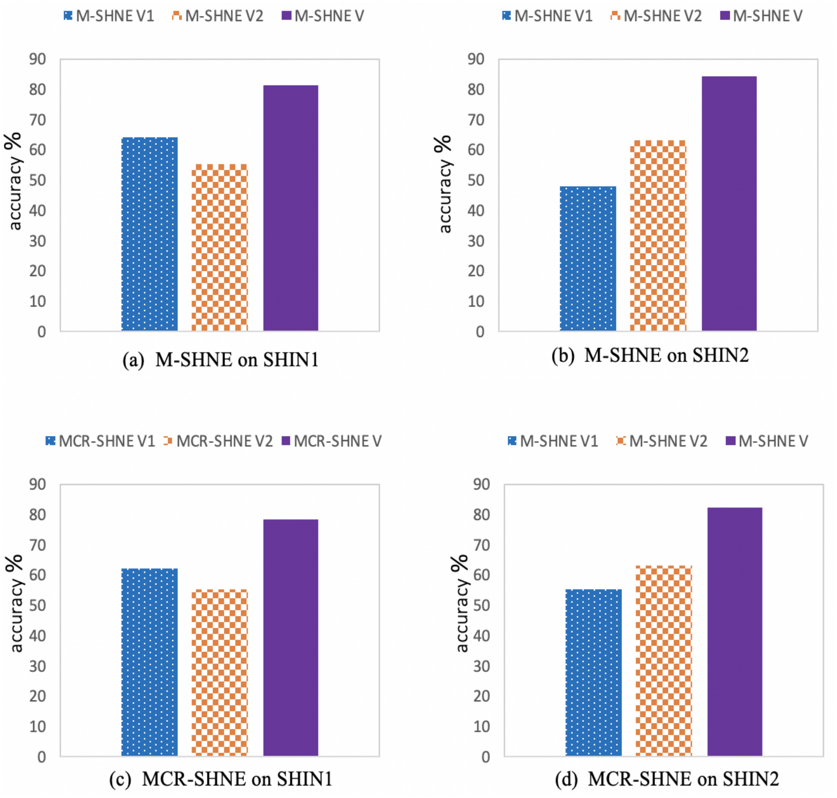
(b) Clustering on SHN2

Figure 5.4. Visualization of clustering results from SHNE

categories within them. We perform k-means clustering on the generated SHNEs of SHN1 and SHN2 from the baseline models and our two developed models. We use the parameter settings mentioned in Section 5.4.3 for generating the embeddings and k is set to 5.

Figure 5.4 shows the clustering representation of communities that share similar functionality and representations learned from SHNEs of all the different models. We observe that the embeddings generated for SHN1 and SHN2 from our two developed models learn better clustering and separation of the vertices, and the boundaries of each group are much clearer when compared to other baseline models. Specifically, the results shown in Figure 5.4(a) and Figure 5.4(b) for SHN1 and SHN2 prove that network embeddings learned from our developed models better preserve spatial and temporal proximity measures when compared to baseline models.

5.4.7. Spatial and Temporal Influence on SHNE. In this section, we conduct experiments to demonstrate the influence of spatial and temporal attributes on the quality of generated SHNE. Here, we create two different versions of networks for both SHN1 and SHN2. Specifically, V1 is a simplified network where spatial attributes associated with all POI pairs in the network are ignored; this eliminates the spatial associations in successive check-in behaviors of users. In V2, we ignore the temporal characteristics of user mobility between POIs so that no specific temporal periodic pattern is utilized in the embedding process. Finally, we use the original network V, which integrates spatial attributes associated with all POI pairs and the temporal characteristics of user mobility between POIs. We compare performances on the task of classification, and accuracy is used as the evaluation metric. We use 65% of data for training, 35% for testing and POI categories are used as labels.



(a) M-SHNE on SHIN1

(b) M-SHNE on SHIN2

(c) MCR-SHNE on SHIN1

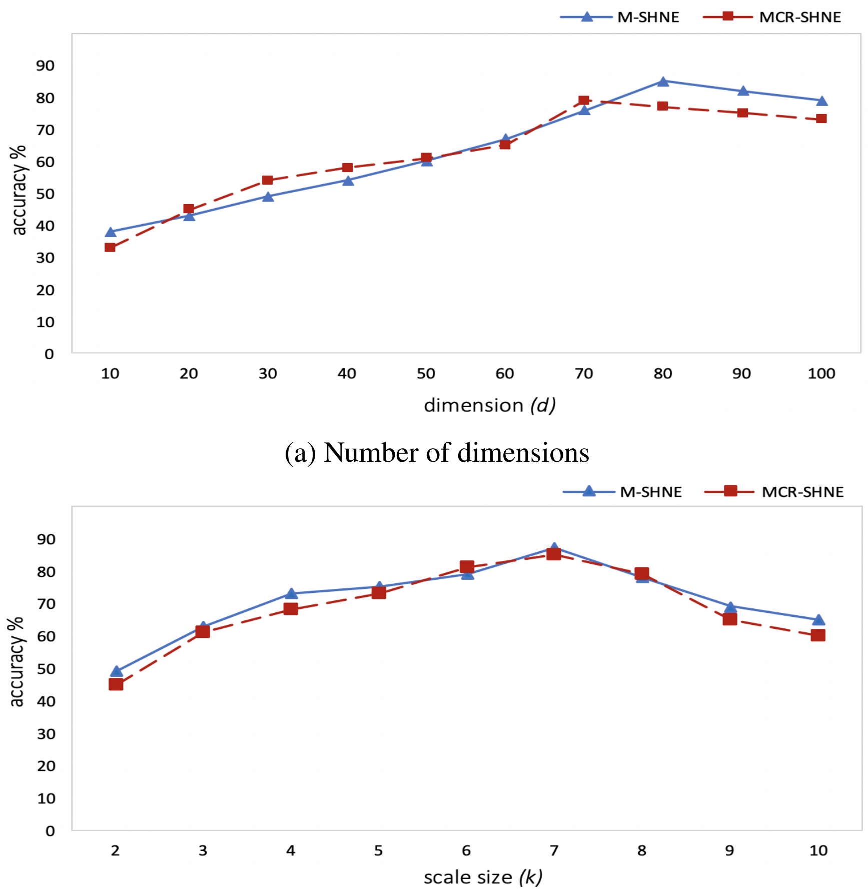
(d) MCR-SHNE on SHIN2

Figure 5.5. Spatial and temporal influence on SHNE

Figure 5.5 shows the results of our experiment. We observe that in cases where we use SHN1, embeddings learned from V1 seems to perform better than embeddings learned from V2. Whereas in the case of SHN2, embeddings of V2 perform better than embeddings of V1. A possible explanation for these observed results could be the fact that SHN1 contains a vast amount of temporal relations generated from human mobility (bike trips) between vertices (POIs); this results in V1 embeddings performing better. In the case of SHN2, since it includes more spatial relations between vertices (POIs), V2 embeddings perform better. However, in all the cases, network embeddings generated from V1 and V2 networks underperform when compared to embeddings generated from network V. Thus, the results indicate that preserving both spatial and temporal relationships within the network helps us generate higher quality embeddings.

5.4.8. Parameter Sensitivity. We discuss the parameter sensitivity factor in this section. Specifically, we assess how the different choices of dimension *d* and maximal scale size *k* can affect the quality of SHN embeddings learned from our models. We measure the quality of our SHN embeddings based on the accuracy achieved in vertex classification and clustering tasks.

Figure 5.6 shows the classification accuracy achieved while using SHN embeddings generated from our two developed models over different settings of dimension *d*. At first, the accuracy shows an apparent increase. This is intuitive as more bits can encode more useful information in the increasing bits. However, when the number of dimensions continuously increases, the performance starts to drop slowly. The result proves that having too small a dimension *d* can be inadequate for capturing rich information of SHNs. However, a larger dimension *d* may also introduce noise which will deteriorate the quality of SHN embedding and reduce the performance accuracy. Hence, it is necessary to find an optimal *d* value for embedding. Figure 5.9 shows the accuracy scores over different choices of *k* on the clustering task. We can observe that the setting *k*=7 has a significant improvement over the setting *k*=2, and *k*=10. The above observation confirms that different k-order can



(a) Number of dimensions

(b) Number of scales

Figure 5.6. Parameter Sensitivity

learn complementary local information. However, when *k* becomes too large or too small, learned k-order relational information becomes too weak to capture the original structural and semantic information of SHNs.

5.5. CONCLUSION

In Section 5, we introduces two comprehensive frameworks for learning SHN embeddings. Both models develop new approaches for efficiently capturing higher-order spatial and temporal interactions of real-world SHNs. To effectively explore potential relationships among vertices M-SHNE leverages random walks and a multilingual word embedding technique to generate a set of multilingual corpora. We then formulate an objective function that collectively maps each lexical vector of multilingual corpora to its semantic space in the low-dimensional space embedding, effectively preserving the intrinsic structure of the SHN. We also develop a second method called MCR-SHNE that combines the advantages of PCRW and path counting algorithm first to construct a corpus consisting of collective meta-path instance that encapsulates the multiplex of complex relations between vertices and edges in a SHN. Word embedding technique is then employed to generate a fused SHN embedding that incorporates innate spatiotemporal relationships between vertices of a SHN.

As shown in our extensive experiments, our network embedding methods can recover the original network more accurately, and have better performances over data mining tasks like classification and clustering over state-of-the-art network embedding methods. Network embeddings generated from our two developed methods can also be adapted to learn real-world urban community structures. They can effectively learn from dynamic human mobility patterns between static geospatial entities of a city's SHN and identify urban communities that share similar functionalities and representations, and distinguish between POIs of different categories, as demonstrated in the experiments.

Analysis of spatiotemporal data has applications in various fields, one of which is political science. Given the availability of sufficient data, the developed framework can be applied to study human mobility patterns and urban communities with respect to factors such as ethnicity, linguistic, socioeconomic status, etc. The insights gained from the framework can be used to detect or create Gerrymandered districts.

6. REPRESENTATION LEARNING FRAMEWORK WITH NODE SENSE DISAMBIGUATION

In this section, a spatiotemporal context-aware network embedding framework that jointly captures the spatial regularities between objects and the sequential transition patterns of human mobility by learning contrastive context senses associated with nodes is presented. The framework first models the heterogeneous urban mobility data collected from multiple sources as an SHN using a probabilistic weighted degree centrality measure. To learn the sequential transition patterns of human mobility in urban regions, meta-path constrained random walks (MPCRWs) are performed on the constructed SHN, which captures the proximities between multi-typed objects via their rich spatiotemporal links. By treating the generated meta-path instances as sentences, the framework captures multiple contrastive context senses associated with nodes in an SHN produced due to a multiplex of spatial and temporal dependencies between objects in urban mobility data; this is facilitated by performing spectral graph clustering. The learned contrastive contextual node senses are mapped with respective meta-path instances. Finally, latent embeddings of the mapped meta-path instances are learned by using the word2vec model skip-gram. The performance of the developed framework is evaluated on real-world application problems. Experimental results demonstrate the effectiveness of the framework over state-of-the-art alternatives.

6.1. BACKGROUND AND OVERVIEW

An SHN of a region contains real-world objects such as places, and things whose interrelationships are defined on a multitude of complex spatial and temporal attributes. Effective analysis and representation of the static spatial relationships between geographical structures and the dynamic nature of human mobility patterns are vital for building many critical applications like designing smart traffic systems, understanding the evolution of urbanization, etc. However, most developed network embedding models struggle to capture efficient representations of static spatial and dynamic temporal relationships between nodes in an SHN. It is challenging to model large-scale SHNs generated from different sources and learn their latent embeddings that encapsulate the context of relationships between real-world objects defined by their spatial and temporal dependencies.

An effective SHN embedding model should be capable of capturing geographic influence as well as temporal influence. Recently, the study of spatiotemporal data from sources such as mobile devices, GPS data from vehicles, and online social networks that include user check-in information has shown that there is a spatial corelation between a user's visit pattern and geographic proximity between places the user visits [73]. For example, users who want to shop usually check-in to places that are close to each other. We believe that a user's sequence of activities is influenced by geographic proximity between places that are previously visited. By studying geographic influence we can gain valuable insights into how geographic proximity between structures influences human mobility patterns. This, in turn, can help us more accurately represent static allocations of various geographical structures. By studying temporal influence we can learn about how temporal factors affect human mobility patterns. For example, people usually maintain a fixed routine of daily check-ins they perform. On weekdays, a user might check-in to locations closer to work. During weekends, the same user might check-in to bars and restaurants in a commercial region. Some work have focused on temporal drifting [74, 75], but they ignore multiple contrastive contextual node senses an individual node can possess. The nature of the relationship between nodes in an SHN modeled over urban mobility data is dynamic due to the multiplex of spatial and temporal dependencies. This results in multiple spatial and temporal contexts that a node can be associated with which in turn can cause node sense ambiguity. This problem is very similar to the lexical ambiguity problem faced in the field of Natural Language Processing (NLP) where the words have more than one meaning given their context of occurrence [76]. Hence, it is essential to learn contrastive contextual node senses for generating effective SHN embeddings. Learning individual node senses can help us better capture the structural similarity between nodes in the latent representation. Figure 6.1 shows an SHN of a region where Points of Interest (POIs) or nodes are interconnected by multiple relationships such as day of visits, POI categories, Check-In frequency, time of visit, etc. Nodes are represented by POIs and the nature of their interconnecting relationships are defined within the box.

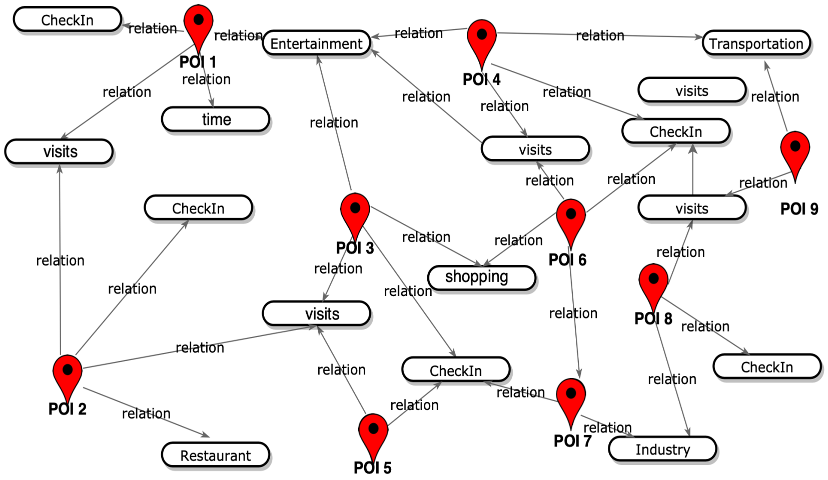


Figure 6.1. Spatiotemporal Heterogeneous Network

The focus of this work was to develop a framework that can collectively learn the network embeddings of an SHN modeled through urban mobility data for the identification and quantification of complex urban structures. In what follows, we outline how we develop a spatiotemporal context-aware network embedding framework that can address:

- How to quantify, unify, and construct an SHN in uniform model space from urban mobility data generated from diverse sources

- What are the sequential transition patterns of human mobility and how are they influenced by spatial and temporal attributes

- How to delineate the spatiotemporal context of a node and assimilate the learned contrastive node senses in the generated network embeddings

The developed framework uses a probabilistic weighted degree centrality measure to quantify a multiplex of relationships between real-world objects and model the SHN. It then employs MPCRWs to capture semantic spatial and temporal aspects of urban mobility patterns from the constructed SHN. The generated MPCRW instances are sequences of structurally similar nodes observed by weighted random walks traversing an SHN. Figure 6.2 shows two meta-path instances between nodes generated from weighted random walks. The framework then utilizes spectral graph clustering to learn contrastive node senses based on meta-path instances. The different contexts are learned based on the occurrences of nodes in multiple meta-path instances. Finally, it leverages the use of skip-gram to learn representations of the nodes. This approach helps us to effectively regularize the model and enhance generalization capability.

We evaluate the performance of our developed embedding framework on the task of recommending relevant POIs for user visits. We also evaluate the quality of the learned embeddings by comparing it against state-of-the-art network embedding methods over data mining tasks of classification and clustering. Experimental results show that the developed method outperforms all alternative approaches in most of the qualitative measures used.

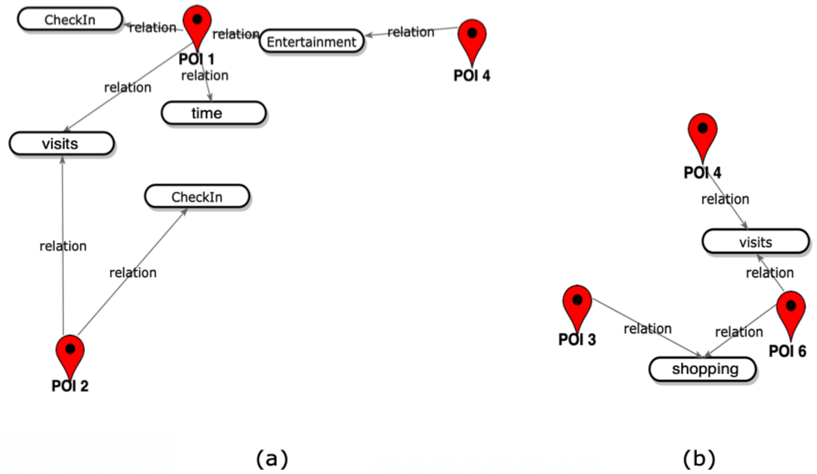


Figure 6.2. Meta-path instances

Overview. The remainder of Section 6 is organized as follows. In Section 6.2, we provide the problem definition for the spatiotemporal heterogeneous network embedding framework. Section 6.3 provides details about the framework of our developed model. In Section 6.4, we report the experimental results of our developed model. Section 6.5 provides a brief review of related works. Finally, we conclude in Section 6.6.

6.2. PROBLEM STATEMENT

In this section, we formalize the problem of spatiotemporal network embedding by providing some key definitions.

Definition 1: *(Point of Interest (POI)):* A POI is a unique geographic structure or a point location into which the users check-in. A POI is represented as: $< p_{id}, longitude, latitude >$. We use the terms POI and node interchangeably.

Definition 2: *(Check-in):* Check-in is a user activity indicator. It is reported by the user u, in location l, at time t. It can be represented as: $c_i =< u, l, t >$. A single user can have multiple check-ins, $u_i =< c_1, \cdots, c_n >$

Definition 3: *(Spatiotemporal Heterogeneous Network):* An SHN represents a multiplex of complex spatial and temporal relationships that exists between real-world objects. Here, we consider the POIs as nodes and human mobility between POIs as weighted links. We can represent an SHN as $G = (V, E, \phi, \psi)$ where $\phi : V \rightarrow \mathcal{L}$ is a mapping function for nodes and $\psi : E \rightarrow \mathcal{R}$ is a link type mapping function. G is a directed graph defined over nodes V and edges E where each node $v \in V$ belongs to an object type $\phi\,(v) \in \mathcal{L}$ and each link $e \in E$ belongs to a link type $\psi\,(e) \in \mathcal{R}$.

Definition 4: *(Meta-Paths):* A set of Meta-paths $\mathcal{P}$ in an SHN can be used to capture various semantic spatial and temporal relationships between nodes. A meta-path instance p_i defined on an SHN $G = (V, E, \phi, \psi)$ can be represented as $v_1 \xrightarrow[e_1]{} v_2 \xrightarrow[e_2]{} v_i \xrightarrow[e_i]{} \cdots \xrightarrow[e_{n-1}]{} v_n$, where v represents nodes and e is the edge that represents composite relationships between nodes.

Figure 6.2 shows different instances of meta-paths. Figure 6.2(a) exhibits a meta-path instance $POI_2 \rightarrow d \rightarrow POI_1 \rightarrow c \rightarrow POI_4$. POI_2 and POI_4 are connected based on day of visit (d) and check-in (c) frequency. The meta-path instance $POI_3 \rightarrow s \rightarrow POI_6 \rightarrow d \rightarrow POI_4$, shown in Figure 6.2(b), indicates that POI_3 is connected to POI_4 via POI_6 where POI_3 and POI_6 belong to the same POI category shopping (s) and POI_6 and POI_4 are connected via the day of visit (d).

Definition 5: *(Contrastive Node Sense):* Given a node in an SHN with multiple spatial and temporal dependencies, it is essential to learn the multiple contexts it is associated with. We define the task of learning contrastive node senses as a clustering problem. The node senses N_s learned by clustering meta-paths are then mapped to respective nodes $N_s\,(v_i)$, such that $N_s\,(v_i) \subseteq senses_D(v_i)$ where, $senses_D(v_i)$ is a set of contrastive node senses for node v_i.

Problem Definition. Formally, given a heterogeneous dataset of an urban region with spatial and temporal attributes, the goal of our problem is to first model the observed mobility patterns as a spatiotemporal heterogeneous network (SHN) $G = (V, E, \phi, \psi)$. We then formulate a function $f : V \rightarrow \mathbb{R}^d$ that maps each node $v \in V$ of the SHN to a vector in d dimensional space $\mathbb{R}^d$, such that the spatial allocations indicating the importance of geographical structures and the dynamic sequential transition of human mobility pattern between the geographical structures are preserved.

6.3. METHODOLOGY

In this section, we describe our developed spatiotemporal context-aware network embedding framework. In Section 6.3.1, we elaborate on how we use the developed probabilistic weighted degree centrality measure to construct the SHN. Then, we explain how MPCRWs are used to generate meta-path instances that represent the sequential transition of human mobility in Section 6.3.2. Section 6.3.3 shows how we leverage the use of spectral graph clustering to learn different contexts from the generated MPCRW instances to build a set of node senses. Finally, Section 6.3.4 describes how we employ skip-gram for learning SHN embeddings of a region.

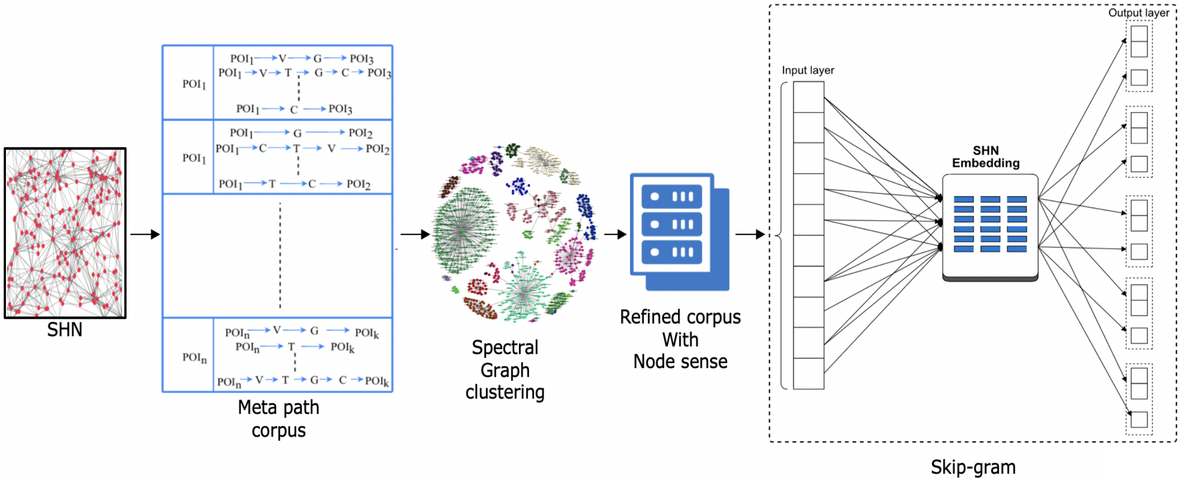

SHN
POI_1
$POI_1 \to V \to G \to POI_3$
$POI_1 \to V \to T \to G \to C \to POI_3$
$POI_1 \to C \to POI_3$
POI_1
$POI_1 \to G \to POI_2$
$POI_1 \to C \to T \to V \to POI_2$
$POI_1 \to T \to C \to POI_2$
POI_n
$POI_n \to V \to G \to POI_k$
$POI_n \to T \to POI_k$
$POI_n \to V \to T \to G \to C \to POI_k$
Meta path corpus
Spectral Graph clustering
Refined corpus With Node sense
Input layer
SHN Embedding
Output layer
Skip-gram

Figure 6.3. Spatiotemporal Context-Aware Network embedding framework

6.3.1. SHN Construction.

Due to the heterogeneous nature of large scale SHNs, it is quite challenging to delineate the complex spatial and temporal relationships that exist between real-world objects. This task becomes even more complex when building SHNs of a region using its urban mobility data. Nodes in an SHN represent important geographic locations/Point of Interests (POIs) and the heterogeneous links represent the multi-class relationships that exist between the nodes. For quantification of a network's interconnectedness strength, a great variety of centrality measures have been developed [52, 53]. Ideally, when building weighted networks we use degree centrality measure for quantification of its interconnectedness strength. Degree centrality measure is one of the simplest measures which uses the number of links between nodes as an indicator of a node's interconnectedness strength. However, in this case, we have to keep in mind the fact that the spatiotemporal data is collected from various sources like GPS devices of vehicles, mobile devices, social media, etc. Also, real-world data are messy, we don't always have details about the exact origin location or the destination location. Hence, to overcome this problem we develop a probabilistic weighted centrality measure that fits spatiotemporal data from various sources and does not require the exact origin POI and destination POI information. We first calculate the probability of a user visiting a POI using $\Delta(x)$:

$$\Delta\left(x\right) = \frac{\beta_1}{\beta_2} \cdot x \cdot exp\left(1 - \frac{x}{\beta_2}\right) \tag{6.1}$$

where, x is the distance between the original drop-off point d and the destination POI. β_1 *is the maximum probability of a user visiting the destination POI from the drop-off point under the function* $\Delta\left(x\right)$*.* β_2 *is the optimal walking distance between the drop-off point and the destination POI with respect to* β_1 *value.* β_1 *and* β_2 *are used as hyperparameters to control the shape of the function* $\Delta\left(x\right)$*.*

We then proceed to calculate the weight of a node based on its diversity and density aspects. We define density as the total number of times a user visits a POI.

$$\delta(p_i) = \sum_{i=1}^{n} \Delta_x \tag{6.2}$$

where, $\delta(p_i)$ *is the density measure of the node* p_i *and* Δ_x *is the number of visits.*

We then calculate the diversity aspects based on the total number of different categories of POIs connected to the node. This measure accounts for the heterogeneity of the network.

$$\gamma(p_i) = 1 - \frac{\sum_{i=1}^{n} P_i(P_i - 1)}{P_c(P_c - 1)} \tag{6.3}$$

where, P_i *is the total number of POIs in individual POI categories and* P_c *is the total number of POI categories.*

The weight of the node is then calculated by fusing the density and diversity aspects of the node.

$$\mathcal{W}_{p_i} = \delta(p_i) \cdot \gamma(p_i) \tag{6.4}$$

where, $\mathcal{W}_{p_i}$ *is the weight of the node* p_i*.*

By considering the density and diversity factors of the node, its significance can be quantified more effectively and also help represent the network structure more accurately.

6.3.2. Meta-path Constrained Random Walks. This section elaborates on meta-path constrained random walks. Random walks have often been employed by many works to learn the structural similarities present within the network. The significance of the node is determined by estimating the likelihood of observing node v_i given all the previous nodes visited so far in the random walk. Meta-path, in general, was developed as a technique through which meta-structure of the information could be explored. It can also be used to capture numerous semantic relationships across multiple entities systematically. As meta-paths carry rich information about the semantic relationships between entities, they can be used to capture the proximity between nodes in a network using several meta-path-based similarity measures.

Given an SHN, a meta-path based proximity between two nodes can be calculated based on meta-path instance p_i as:

$$\sigma\left(v_s, v_t\right) = \sum s\left(v_s, v_t | \mathcal{P}\right) \tag{6.5}$$

where, $\sigma\left(v_s, v_t\right)$ *is the proximity score between nodes* v_s *to* v_t*.*
This is calculated based on the probability of a PCRW restricted on a meta-path following the instance $p_{iv_s \to v_t}$. The probability score between a pair of nodes is calculated $v_s, v_t \in V$ based on the meta-path instances generated from PCRWs as:

$$s\left(v_s, v_t | \mathcal{P}\right) = \sum_{p_{v_s \to v_t} \in \mathcal{P}} s\left(v_s, v_t | p_{v_s \to v_t}\right) \tag{6.6}$$

where, $s\left(v_s, v_t | p_i v_s \longrightarrow v_t\right)$ *is the proximity score w.r.t meta-path instance* $p_{iv_s \to v_t}$ *between* $\left(v_s, v_t\right)$*.*

Table 6.1 lists some meta-path instances from Figure 6.1 and gives insight into the relationships between nodes and their respective semantic meanings. Equation 6.5, is used

Table 6.1. Meta-paths between POIs in SHN of Figure 6.1

Meta-path	**Semantic Meanings**
$POI_1 d \rightarrow POI_2$	POI_1 and POI_2 are visited on the same day (d)
$POI_3 \rightarrow e \rightarrow d \rightarrow POI_4$	POI_3 and POI_4 belong to the POI category entertainment (e) and are visited on same day (d)
$POI_4 \rightarrow c \rightarrow d \rightarrow POI_8$	POI_4 and POI_8 on average have similar check-in (c) frequency and are visited on the same day (d)
$POI_5 \rightarrow c \rightarrow POI_7 \rightarrow i \rightarrow POI_8$	POI_5 and POI_8 are connected via POI_7. POI_5 and POI_7 have similar check-in (c) frequency, POI_7 and POI_8 belong to the same POI category industry (i)

to calculate the proximity between two nodes as the sum of proximity scores of all the meta-paths. Shorter meta-paths tend to be more informative than the longer ones. Meta-paths with longer lengths are ineffective when indicating the relationship between nodes. The number of possible meta-paths grows exponentially based on the length of meta-paths. It can even become infeasible in certain large scale information networks. To overcome this, we restrict the length of the meta-path using a threshold l.

$$\hat{\delta}_l\left(v_s, v_t\right) = \sum_{len(\acute{p}) \leq l} s\left(v_s, v_t | \mathcal{P}\right) \tag{6.7}$$

We can calculate the proximity based on path constrained random walks as:

$$\hat{\delta}_l\left(v_i, v_j\right) = \sum_{(v_i, v') \in E} \acute{p} v_i \longrightarrow v'^{\psi(v_i, v')} \cdot \hat{\delta}_{l-1}\left(v', v_j\right) \tag{6.8}$$

where, $\acute{p} v_i \longrightarrow v'^{\psi(v_i, v')}$ is the probability of a random walk occurring from v_i to v' w.r.t. to the edge $\psi(v_i, v')$.

6.3.3. Learning Contrastive Node Senses. Given the dynamic nature of spatial and temporal relationships between nodes in an SHN, it is essential to learn their contrastive node senses for generating effective node embeddings. Each sense of a node in an SHN corresponds with a particular kind of context it appears in, and the similarity between node senses can be measured by their corresponding contexts. This problem becomes more convoluted when attempting to learn the node sense of a node that represents a POI, such as a Mall that offers multiple services as it incorporates restaurants, theaters, departmental stores, clothing stores, tech stores, etc. In our model, we use spectral graph clustering to cluster the multiplex of meta-path instances such that meta-path instances that refer to the same context appear in the same cluster. By clustering the meta-path instances, we can learn different contextual node senses of each node based on the multiple meta-path instances that they occur in. By learning contrastive node senses we can more accurately interpret the different contexts in which they appear.

Spectral clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them. Spectral clustering uses information from the eigenvalues (spectrum) of special matrices built from the graph or the data set. Given the MPCRW based corpus, a spectral graph clustering algorithm can always generate a division, no matter whether the structure exists or not.

Following the mathematical procedure, we first define the Laplacian L as:

$$L = D - A \tag{6.9}$$

where, Dis the diagonal matrix and A is the adjacency matrix.

We then define the normalized Laplacian as:

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{6.10}$$

Eigenvalues and eigenvectors of L are then calculated. The smallest eigenvalue λ_0 is always equal to zero. The multiplicity of zero among the eigenvalues is equal to the number of connected components in the graph. We consider the eigenvector belonging to the second smallest eigenvalue λ_1, also called *Fiedler value* and *Fiedler vector*, that correspond to the nodes in the network. We then sort the *Fiedler vector*, thus sorting the nodes in the graph. We then make $n - 1$ cuts along the *Fiedler vector*, dividing the elements of the vector into two sets. We then compute the conductance for each cut as:

$$\phi\left(S\right) = d\left(V\right)\frac{\left|\partial\left(S, \bar{S}\right)\right|}{d\left(S\right), d\left(\bar{S}\right)} \tag{6.11}$$

where, $d\left(S\right) = \sum_{i \in S} \mathcal{W}_{p_i}$. *Total weight of the edges between* S *and* $\bar{S}$ *is indicated by* $\partial\left(S, \bar{S}\right)$. $V = S + \bar{S}$ *is the set of all nodes in the graph.*

We then choose the cut with the smallest conductance $\phi\left(S\right)$, by deleting the edges between S and $\bar{S}$. The procedure is carried out until the conductance $\phi\left(S\right)$ reaches threshold t.

The task of embedding node sense by using the MPCRW corpus as an inventory of nodes is achieved by clustering MPCRW instances together based on vector similarities. The centroids of the clusters are inferred as representing node senses.

$$N_s\left(v_i\right) = \overset{argmin}{\underset{v_i \in V}{}} d\left(s_i, v_i\right) \tag{6.12}$$

where, $N_s\left(v_i\right)$ *is node sense of nodes* v_i *and* s_i *is the proximity score of MPCRW.*

6.3.4. SHN Embedding. Analysis of the SHN dataset consisting of POIs and the user's check-in data showed that the frequency of POI check-ins follows the power-law distribution similar to word frequency distribution. This aspect motivated us to use the famous word2vec model skip-gram [11] to learn sequential transition of human mobility between POIs. Figure 6.3 shows the framework of our developed model and Algorithm 4 shows the steps for generating the SHN embeddings.

Algorithm 4 SHN Embedding

```
Input: graph $G = (V, E, \phi, \psi)$
       window size $\omega$
       embedding size $d$
       walks per node $\gamma$
       path constraint $C$
       walk length $t$
       meta-path length $l$
Output: matrix of node representation $\Phi \in \mathbb{R}^{|V| \times d}$

for each $v_s \in G$ do
    for $i = 0$ to $\gamma$ do
        Rlength = 0
         Mlength = 0
         if $Rlength < t \;\&\&\; Mlength < l$ then
             $p_{v_s \to v_t}$ = select $v_t$ with prob $\frac{1}{|v_t \in C|}$
              $\mathcal{M}_v \leftarrow p_{v_s \to v_t} + 1$
              $Mlength \leftarrow Mlength + 1$
              $Rlength \leftarrow Rlength + 1$
        end
    end
end
compute laplacian matrices $\left\{\mathcal{L}_t^{(p)}\right\}$
compute first $k$ eigenvectors $u_1 \cdot \cdot \cdot \cdot u_k$ of $\mathcal{L}$
for each $v_j \in \mathcal{M}_v$ do
    assign $v_j \longrightarrow$ cluster $c_k$
end
for each $v_j \in \mathcal{M}_v$ do
    $N_s\left(v_j\right) = c_1 \cdot \cdot \cdot \cdot c_k \longrightarrow v_j$
end
for each $v_j \in \mathcal{M}_v$ do
    $\mathcal{W}_{v_i} = \mathcal{M}_v \longrightarrow \mathcal{M}_v + 1$
    for each $u_k \in W_{v_i}\,[j - w : j + w]$ do
        $J(\Phi) = -logPr\left(u_k|\Phi\left(v_j\right)\right)$
        $\Phi = \Phi - \alpha * \frac{\partial j}{\partial \Phi}$
    end
end
```

We use the truncated meta-path constrained random walk instances $\acute{p}$ of length l along with path constraint factor C to control the length and the direction of the random walks. We utilize the two hyper-parameters to generate truncated MPCRW instances that effectively capture semantic structural information between nodes. Shorter MPCRW instances are more informative than the longer ones as longer meta-paths link more remote nodes that might be less contextually related [14]. The generated meta-path instances are then infused with node senses that are based on the different contexts than they occur in using spectral graph clustering.

The goal of the developed spatiotemporal network embedding framework is to create an SHN embedding using meta-path based corpus given a SHN $G = (V, E)$. To achieve this, we leverage the use of skip-gram. The corpus comprised of MPCRW instances infused with node senses is then passed as input to skip-gram. The skip-gram then iterates over all possible collocations of MPCRW instances that appear within the window w of the corpus $\mathcal{M}_v$. It maps each node v_j to its current representation vector $\Phi(v_i) \in \mathbb{R}^d$ and generates SHN embeddings that preserve complex spatial and temporal proximity measures between nodes in an SHN.

We also employ negative sampling while learning the SHN embeddings using skip-gram. Negative sampling [11] is a loss function which rewards the estimate of the probability for nodes that co-occur with each other, while penalizing the estimate of the probability for random node pairs co-occurring with each other. Negative sampling has proven to be a very effective alternative to the computationally expensive softmax, where we need to sum the overall non-zero proximity scores $s(v_i, v_j)$ for a specific node v_i. Negative sampling has achieved state of the art results in many Natural Language Processing (NLP) tasks and plays a major role in substantially speeding up the learning process and helps generate better embeddings [72]. Equation 6.13 describes the loss function for each node context occurrence in a SHN corpus generated from our developed model. Here, we consider $v \in V$ as the target node, $c \in C$ as a context node and $c_n \in C$ as a random negative sample (indexed by n) drawn from a noise distribution $p\left(c\right)$.

$$J_{sg}\left(v, c\right) = \log\left(\sigma\left(c^t v\right)\right) + \sum_{n=1}^{N} \mathbb{E}_{c_n \sim P(c)}\left[\log\left(\sigma\left(-c_n^t v\right)\right)\right] \tag{6.13}$$

The total loss is the summation of Equation 6.13 for all pairs of nodes v and c co-occurring in the corpus, as extracted using a context window of size w.

6.4. EXPERIMENTAL SECTION

This section details the empirical evaluation of the developed framework assessed on an SHN modeled on real-world urban mobility data procured from multiple sources.

6.4.1. Data Description and Analysis. In this section, we provide details of the three data sets used for constructing the SHN. Table 6.2 shows the statistics. The Mobike dataset was released as a part of the Mobike big data challenge in 2017 by Mobike. The data set period ranges from May-10 to May-24 of 2017. It contains details of 3,214,096 trips and is comprised of attributes like userId, orderId, trip start time, trip start location, and trip end location. The POI data set for the city of Beijing was obtained from www.dianping.com; it is a commercial review and recommendation website. It consists of 328,668 POIs divided into 20 different categories like Hospitals, Malls, Restaurants, Theaters, etc. The Weibo & Jeipang are Chinese social media data sets that together include 2,020,967 check-in entries

Table 6.2. Statistics of the datasets.

Data Sources	**Attributes**	**Statistics**
Mobike Trips	# trip records # users Trip start time Trip start location Trip end location Bike id	3,214,096 349,693
	Time period of records	05/10/2017 - 05/24/2017
POIs	Number of POIs # POI categories	328,668 20
Weibo & Jiepang	# Check-In # users POI Name POI Location POI address	2,020,967 212,362

of POIs in Beijing. They contain attributes like POI name, POI check-in time, POI address, and POI <*lat*, *long*> location.

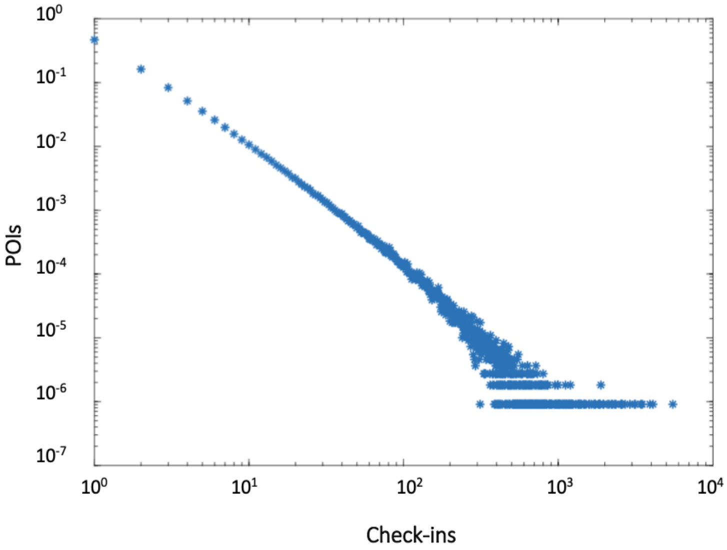


Figure 6.4. Distribution of user visits

Initial analysis of the three data sets revealed that the user check-in activity follows the power-law distribution. It gives us an insight into how some patterns cover a wide range of magnitudes. The power-law predicts that most nodes in the network have a few connections, while a small number of nodes, have a rich local neighborhood. Accordingly, Figure 6.4 shows that only a few popular POIs have high check-in activity whereas in the majority of POIs the number of check-ins are close to or below the mean value.

6.4.2. Modeling an SHN Over Urban Mobility Data. To demonstrate the effectiveness of our model on very large and complex SHNs, we construct an SHN by combining Mobike, POI, and Weibo & Jeipang datasets. We initialize 1,765,025 trip start and end locations with visits greater than 300 as nodes. We then integrated the POI and Weibo & Jeipang datasets with the existing 1,765,025 nodes using our developed probabilistic weighted degree centrality measure to supplement the heterogeneous spatial and temporal relationship that exists between nodes. Specifically, the constructed SHN encapsulates multi-class spatial and temporal relationships like visits per day, visits per hour, geographic distance, number of individual user visits, high activity time, low activity time, 20 different POI categories, and their associated hierarchical relationships.

6.4.3. POI Recommendation. We evaluate the effectiveness of our developed model based on the task of recommending POIs to users. We also test the superiority of the latent features (LF) learned from our model when compared to the explicit features (EF) highlighted in Section 6.4.2 extracted directly from the dataset.

6.4.3.1. Baseline algorithms. We chose the following Learning to Rank (LTR) models as baselines for comparison.

- RankNet (RN) [63]: This approach uses a feedforward neural network to model the underlying probabilistic cost function.

- RankBoost (RB) [62]: The idea behind RankBoost is to model the learning to rank as a binary classification problem on instance pairs. RankBoost forms the final ranking function by first training one weak ranker at each iteration and then combining these weak rankers.

- ListNet (LN) [59]: This model defines a listwise loss probabilistic function that involves two probabilities called permutation probability and top one probability.

- LambdaMART (LM) [77]: This approach is a boosted version of LambdaRank [78]. It solves ranking tasks by leveraging gradient boosted decision trees using a cost function based on LambdaRank.

- AdaRank (AR) [58]: This method uses a linear combination of weak rankers by performing re-weighted training and then linearly combines all the weak rankers for making predictions.

- Random Forest (RF) [60]: Random Forest learns the ranking of objects by learning the prediction from an ensemble of random trees.

6.4.3.2. Parameter settings. We use the RankLib [79] for implementing the 6 LTR models. For RankNet we set number of epochs = 100, hidden layers = 1, number of hidden nodes per layer = 10, and the learning rate = 0.0005. For RankBoost, we set the number of epochs = 300 and number of threshold candidates = 10. We set number of hidden layers = 5, epochs = 100, number of hidden nodes per layer = 10, and the learning rate = 0.0005 for ListNet. For LambdaMART, we set number of trees = 500, number of leaves = 10, learning rate = 0.01, threshold candidates = 200. For AdaRank, we set the number of iterations = 500, tolerance = 0.002, and feature sampling = 5. For Random Forests, we set number of bags = 300, sub-sampling rate = 1.0, feature sampling = 0.3, learning rate = 0.01. We use 65% of the data for training and remaining 35% is used for testing.

6.4.3.3. Evaluation metrics. We use the following measures for evaluating the performance of the LTR models.

- Normalized Discounted Cumulative Gain (NDCG): NDCG is calculated based on Discounted Cumulative Gain (DCG). DCG is given by,

  $$DCG\,[n] = \begin{cases} rk_{rel} \text{ if } n = 1 \\ DCG\,[n-1] + \frac{rk_{rel}}{log_2 n} \text{ if } n \geq 2 \end{cases}$$

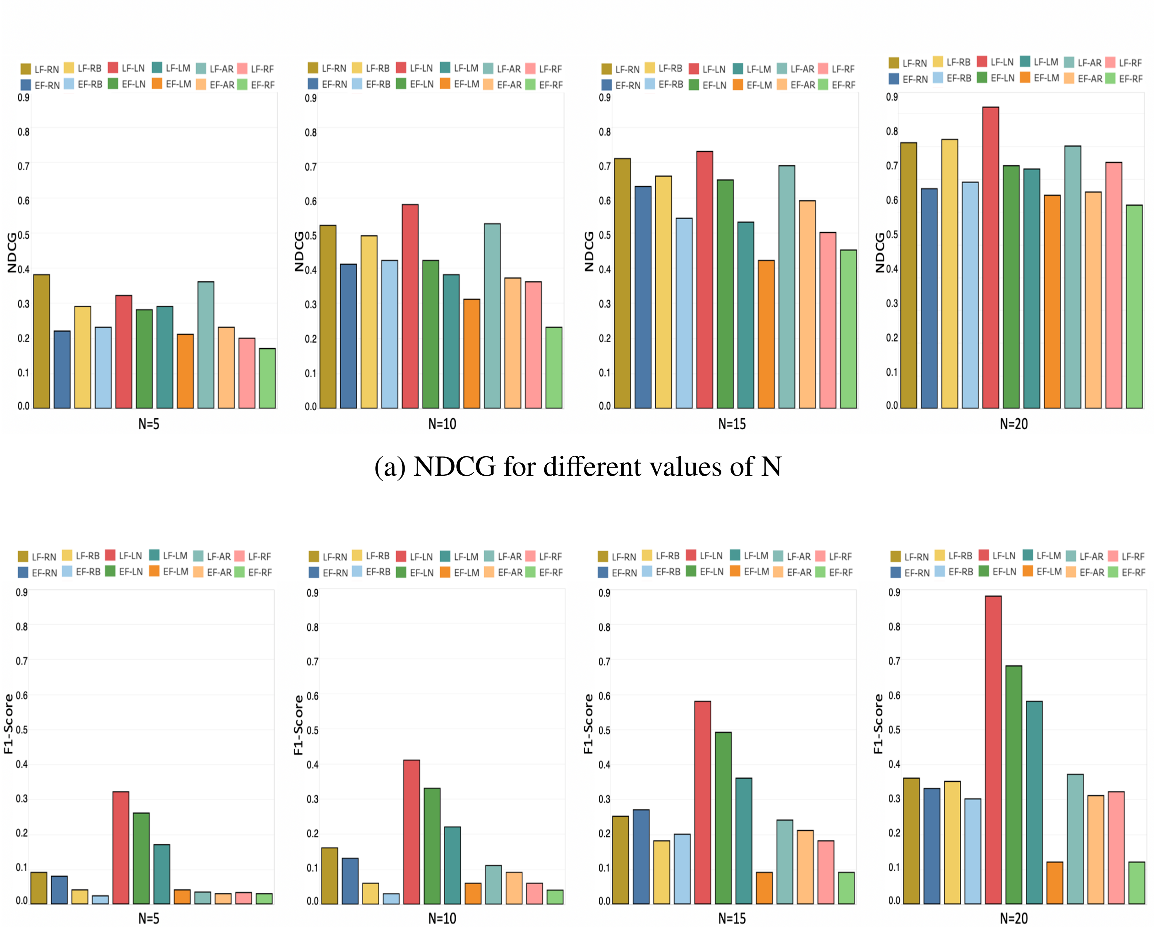
  Given the ideal DCG, we can calculate NDCG at the n^{th} position as $NDCG\,[n] = \frac{DCG[n]}{DCG'[n]}$. The larger the NDCG score is, higher is the top-n ranking accuracy.

- F1-Score: F1-Score considers both precision and recall in a single metric. It is calculated as $F1 = 2\frac{precision \times recall}{precison + recall}$.

6.4.3.4. Quantifying ranking relevance of POIs. The ranking relevancy of POI is based on Equation 6.14. We quantify it based on 3 factors: (i) distance between user's current location and destination POI location; (ii) frequency of user Check-Ins in a given time window; (iii) weight of the node (POI) in the constructed SHN. Due to the sparsity in urban mobility data, most users do not have recent check-in history. For such users, we only consider factors (i) and (iii) to quantify the ranking relevance of a POI visit.

$$\mathcal{Q}(u) = \begin{cases} \mathcal{W}_{p_i} + \|\lambda - d\|^{-1} + \sum_{i=1}^{n} V_{[t-\tau]} & \text{if } V \geq 1 \\ \mathcal{W}_{p_i} + \|\lambda - d\|^{-1} & \text{if } V = 0 \end{cases} \tag{6.14}$$

where, $\mathcal{W}_{p_i}$ is the weight of the node, d is the distance between user's current location and destination POI, λ is the robust mean of distance between user's current location and surrounding POIs, V is the total number of user check-in between time t and time window τ.



(a) NDCG for different values of N

(b) F1-Score for different values of N

Figure 6.5. Overall performance comparisons of the LTR models on latent and explicit features in terms of NDCG and F1-Score.

6.4.3.5. Results and analysis. Figure 6.5(a) and Figure 6.5(b) show the NDCG and F1-Score of the 6 baseline models for different values of N on both the EF and LF learned from the developed method. In all baseline models, we can observe significant improvements in performance when trained on latent features. This shows the effectiveness of latent features over explicit features.

6.4.4. Embedding Evaluation. The developed framework maps the SHN features to latent high-dimensional semantic feature space. The learned semantic feature directly determines the semantic richness of the latent feature. We compare the quality of the embeddings learned from our model against state-of-the-art network embedding models on the tasks of classification and clustering.

6.4.4.1. Baseline models. We chose the following three network embedding models as baselines.

- DeepWalk [9]: DeepWalk uses short truncated random walks to transform a network into node sequences. It then leverages the use of skip-gram to learn its representations.
- LINE [13]: LINE models node co-occurrence probability and node conditional probability of a network. By doing so, it can preserve both the first-order and the second-order proximities of the undirected network.
- Node2vec [10]: Node2vec can strike a balance between local and global properties of a network by leveraging the use of biased random walks for exploring the neighborhood of a node.

6.4.4.2. Parameter setting. We implemented the baseline models using the OpenNE Python toolkit. For DeepWalk and node2vec, we set the number of walks per node = 10, length of random walk = 20, and window size = 10, respectively. For LINE, we set the number of negative samples = 5 and order = 3.

Table 6.3. Accuracy (%) of POI classification on SHN

	% Labeled Nodes	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Macro-F1%	DeepWalk	37.51	33.24	39.53	40.39	41.63	42.17	42.82	44.12	45.08
	LINE	26.75	27.14	28.82	29.04	30.61	31.49	32.23	33.98	35.67
	node2vec	**38.15**	**40.42**	34.07	35.61	36.19	37.82	39.56	41.33	42.71
	ns2vec	36.35	39.81	**43.91**	**45.72**	**46.38**	**47.34**	**49.05**	**49.62**	**50.16**
Micro-F1%	DeepWalk	63.14	67.01	65.52	69.25	70.42	72.03	73.31	74.85	75.47
	LINE	61.56	61.98	62.74	63.62	64.07	65.37	66.42	67.16	68.82
	node2vec	**67.22**	**65.09**	68.63	66.51	67.78	68.04	68.85	69.16	69.94
	ns2vec	65.73	64.87	**70.81**	**71.43**	**73.18**	**74.62**	**75.84**	**76.36**	**77.31**

For our developed model (ns2vec), we set the window size = 10, walk length = 20, walks per node = 10, and negative sample per input sample = 5. We set the embedding dimension d = 300. We use Macro-F1 and Micro-F1 scores as evaluation metrics as there is a class imbalance in the dataset. The number of POIs for each POI category is not equally distributed.

6.4.4.3. Classification. In network analysis, classification is an important task. We performed multi-label classification using a logistic regression classifier. For the task of classification, we used learned embedding as node features, and the POI categories as node labels. Also, to evaluate how the model deals with data sparsity, we performed classification with varying percentages of labeled nodes. Table 6.3 shows classification accuracies with different training ratios on different datasets. The best results are bold-faced.

We observe that our developed model (ns2vec) outperforms the three baseline models in most of the occasions. However, node2vec performs slightly better than ns2vec on two occasions where there was a low number of labeled nodes available for our model to build corpus on and generate quality embeddings. Overall performance also proves that our developed model can better address the data sparsity problem as it consistently outperforms other baselines for varying ratios of labeled nodes.

6.4.4.4. Clustering. We use clustering as a way to group POIs into different groups. Owing to the latent semantics of the generated embeddings, the POIs with the same functional semantics will be more close in the latent semantic space. We perform cosine distance based k-means clustering to verify the discriminability and validity of the latent semantic features on the generated embeddings of the baseline models and our developed model. Here, k is set to 5. Figure 6.6 shows the results of the clustering task. We have used t-SNE to reduce the high dimensional features vector to 2 dimensions. By using the 2 dimensions as x,y coordinates, the nodes can be plotted. We observe that the embeddings generated from our developed model helps learn better clustering, separation of the vertices, and boundaries of each group when compared to the baseline models. The results further demonstrate that network embedding learned from our model can help preserve spatial and temporal proximities more efficiently.

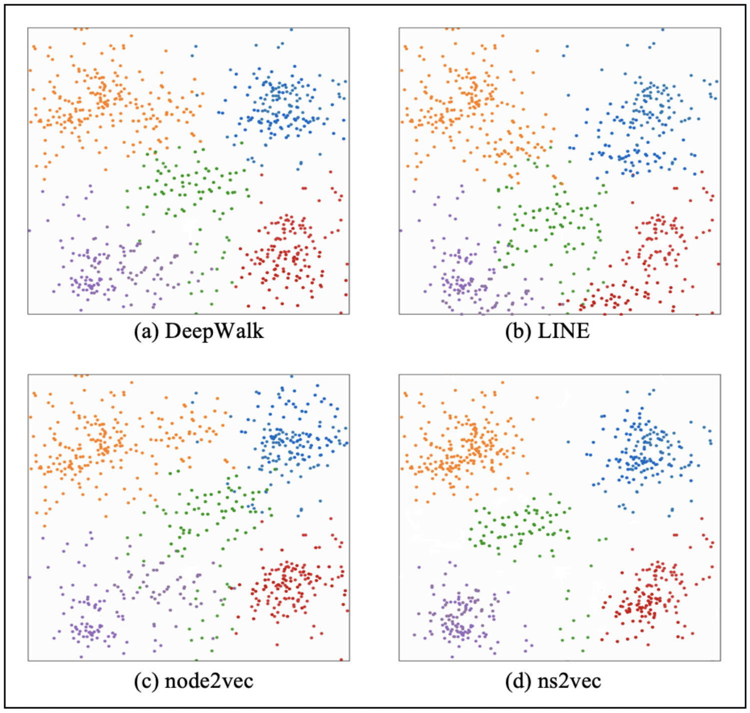

(a) DeepWalk
(b) LINE
(c) node2vec
(d) ns2vec

Figure 6.6. Clustering visualization

6.5. CONCLUSION

Section 6 develops a spatiotemporal network embedding framework that considers the users' current location, the influence of the temporal dimension, and geographic influence while capturing latent representations of the spatiotemporal networks. In particular, the model uses a weighted probabilistic measure to effectively represent the importance of POIs as nodes and sequential transition patterns of human mobility between nodes. It then employs meta-path constrained random walks to learn the structural similarities present within the constructed spatiotemporal network observed by weighted random walks. Also, it captures contrastive node senses based on mutiplex of spatial and temporal contexts that a node in an SHN can be associated with by leveraging the use of spectral graph clustering. For embedding the spatiotemporal network, the model employs skip-gram to map node vectors to their semantic space in the low-dimensional space, effectively preserving the intrinsic structure of the SHN. The effectiveness of learned embeddings is evaluated by first testing it on the task of recommending POIs to users by using learning to rank models. To test the ability of the developed network embedding method to recover the original network more accurately, it is compared against state-of-the-art network embedding methods over the data mining tasks of classification and clustering. The experimental results are validated by comparing them with real-world cases. The POI recommendations results are validated by checking if the user actually visited any one of the n recommended POIs given his location l at time t in real life. The classification and clustering results are validated by using POI categories as true labels. The results prove that the developed framework can be effectively used to capture the relationship between spatial structures and the dynamic sequential transition patterns of human mobility in an urban region.

7. SUMMARY AND CONCLUSIONS

The dissertation systematically discusses the novel frameworks developed to effectively learn the representations of a multiplex of spatial and temporal relationships that exist between nodes in Spatiotemporal Heterogeneous Networks (SHNs). We can summarize their contribution, limitations, and evolution from one framework to the next as follows:

The collective representation learning framework with features importance introduces a collective network embedding approach that uses autoencoders and Laplacian score to learn effective embeddings of SHNs of urban communities [80]. Additionally, it also introduces a new weighted degree centrality measure to more accurately ascertain the node's interconnectedness strength. Experimental results show that the embeddings learned from the developed framework can capture the intrinsic structure of the urban communities more accurately, and outperforms state-of-the-art alternatives. However, the techniques used by the framework to learn the embeddings from constructed periodic SHNs render it inefficient for learning the multimodal correlation that exists between objects in an SHN. This makes the embeddings learned from the framework unsuitable for spatiotemporal applications geared towards traffic and smart transportation systems. This drawback is addressed by the Modality Aware Representation Learning Framework.

The modality aware representation learning framework [81] approaches the problem of learning the multimodal correlation between objects in an SHN by using the tensor factorization method. It first models the spatiotemporal data as multi-dimensional tensors. It then applies the CP decomposition technique to factorize the tensor into a sum of lower-dimensional rank one tensors. Additionally, the tensor factorization method when used to reconstruct the compressed tensor can also predict missing data. The framework leverages these properties to address problems related to traffic and transportation. Specifically, we use the framework to identify regions with demand and supply shortages of stationless bicycle and to address the traffic congestion problems caused due to parked bicycles. The experimental results prove that the developed framework provides better accuracy while reconstructing the original tensor from the factorized tensor, and can also keep the number of parameters to be stored at minimal compared to other baseline models. However, one of the major disadvantages of the developed framework is its inability to capture the intrinsic characteristics of the spatiotemporal features like correlation or mutual information which in turn results in loss of information about spatial and temporal allocations and the significance of geographical structures. This hinders its ability to efficiently learn human mobility patterns and raises the need to build an intuitive framework. Also, the performance of the tensor factorization model in terms of predicting missing data goes down when handling very high dimensional tensors and becomes computationally expensive. These disadvantages are addressed by the Distributed Representation Learning Framework.

To address the problem of learning representations of human mobility patterns encompassed by SHNs, we develop two types of distributed representation learning frameworks [82] that leverage the power-law distribution associated with human mobility patterns . (i) Multilingual SHNE (M-SHNE): It leverages the use of random walks along with a multilingual word embedding technique used in natural language processing (NLP) to collectively learn the spatiotemporal proximity measures of human mobility patterns between nodes in Spatiotemporal Heterogeneous Networks (SHNs) and preserve it in a low dimensional vector space. (ii) Meta-path Constrained Random walk SHNE (MCR-SHNE): It combines the advantage of meta-path counting algorithm, path constrained random walks, and a word embedding technique to generate lower dimensional embeddings that preserve the spatiotemporal proximity measures of human mobility patterns in SHNs. Experimental results demonstrate the effectiveness of the two developed models over state-of-the-art algorithms on real-world datasets. However, as both the developed models use skip-gram for embedding the SHN they suffer from node sense ambiguation problem, i.e. given a node that occurs in multiple different contexts of human mobility pattern the skip-gram is unable to deal with it, resulting in the learning of ambiguous representations. This problem is especially challenging to deal with when attempting to learn the representations of SHNs as most of the nodes appear in multiple contexts and unlike NLP-based algorithms, we cannot resolve the problem by using a supervised labeling approach. This problem is addressed in the developed Node Sense Disambiguation Framework.

The state of relationships between nodes in an SHN is everchanging with respect to spatial and temporal contexts. Thus, to deal with the node sense ambiguation problem posed by the skip-gram model, we develop a node sense disambiguation framework that captures the representations of such nodes by learning their contrastive contextual node senses. Specifically, we use Spectral graph clustering to first calculate the edge weight between node vectors, and then the conductance for each cut is measured. Similar node vectors are then clustered into groups, each identifying a sense of the target node. Each node sense is then mapped to their respective nodes. The new vectors with mapped node senses are then trained using the skip-gram model to create SHN embeddings that incorporate contrastive contextual node senses. Experimental results demonstrate the effectiveness of the node sense disambiguation framework over the developed M-SHNE, MCR-SHNE, and other baseline models. However, the utilization of Spectral Graph Clustering to learn contrastive node senses does make the framework more resource and memory intensive compared to the other two algorithms.

Based on the developed representation learning frameworks for SHNs, possible research direction moving forward could be on developing models that are more interpretable. Specifically, majority of the existing representation learning models lack interpretability. This is due to non-intuitive mappings from data features to salient properties of the representation. Some possible ways to overcome this would be (i) developing visualization tools that can help visualize the modeling process; (ii) investigating the causality between the input and the output.

Additionally, research in the direction of protecting the privacy & security of the data can be valuable. Rich Spatiotemporal data collected from various sources, if not protected, can pose serious privacy and security threats. Therefore, it is essential to handle data of such nature with great care. We can alleviate risks associated by adapting techniques such as federated learning. It offers a way to preserve user privacy by decentralizing data from the central server to end devices.

REFERENCES

[1] Yanjie Fu, Guannan Liu, Spiros Papadimitriou, Hui Xiong, Yong Ge, Hengshu Zhu, and Chen 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.

[2] Yanjie Fu, Yong Ge, Yu Zheng, Zijun Yao, Yanchi Liu, Hui Xiong, and Jing Yuan. Sparse real estate ranking with online user reviews and offline moving behaviors. In *2014 IEEE International Conference on Data Mining*, pages 120–129. IEEE, 2014.

[3] Yanjie Fu, Hui Xiong, Yong Ge, Zijun Yao, Yu Zheng, and Zhi-Hua Zhou. 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*, pages 1047–1056. ACM, 2014.

[4] Yanjie Fu, Hui Xiong, Yong Ge, Yu Zheng, Zijun Yao, and Zhi-Hua Zhou. Modeling of geographic dependencies for real estate ranking. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(1):11, 2016.

[5] Wolfgang Woess. *Random walks on infinite graphs and groups*, volume 138. Cambridge university press, 2000.

[6] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[7] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[8] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[10] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[12] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[13] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[14] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[15] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 121–128. IEEE, 2011.

[16] Ni Lao and William W Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[17] Ni Lao, Tom Mitchell, and William W Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

[18] Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 754–764. International World Wide Web Conferences Steering Committee, 2015.

[19] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[21] Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, 2015.

[22] Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.

[23] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[24] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.

[25] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372, 2013.

[26] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *Twenty-ninth AAAI conference on Artificial Intelligence*, 2015.

[27] Pavlos Kefalas and Yannis Manolopoulos. A time-aware spatio-textual recommender system. *Expert Systems with Applications*, 78:396–406, 2017.

[28] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[30] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, 2008.

[31] Daniel Chemla, Frédéric Meunier, and Roberto Wolfler Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146, 2013.

[32] Hideaki Misawa, Kazuhito Takenaka, Tomoya Sugihara, Hailong Liu, Tadahiro Taniguchi, and Takashi Bando. Prediction of driving behavior based on sequence to sequence model with parametric bias. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[33] Ahmadreza Faghih-Imani and Naveen Eluru. Analysing bicycle-sharing system user destination choice preferences: Chicago's divvy system. *Journal of Transport Geography*, 44:53–64, 2015.

[34] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 841–852. ACM, 2016.

[35] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 165–178. ACM, 2016.

[36] Jiawei Zhang, Xiao Pan, Moyin Li, and S Yu Philip. Bicycle-sharing system analysis and trip prediction. In *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, volume 1, pages 174–179. IEEE, 2016.

[37] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *AAAI*, pages 723–729, 2015.

[38] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, 2016.

[39] Luigi Dell'Olio MEng et al. Implementing bike-sharing systems. *Proceedings of the Institution of Civil Engineers*, 164(2):89, 2011.

[40] Jenn-Rong Lin and Ta-Hui Yang. Strategic design of public bicycle sharing systems with service level constraints. *Transportation research part E: logistics and transportation review*, 47(2):284–294, 2011.

[41] Leonardo Caggiani and Michele Ottomanelli. A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences*, 54:675–684, 2012.

[42] Jon Froehlich, Joachim Neumann, Nuria Oliver, et al. Sensing and predicting the pulse of the city through shared bicycling. In *IJCAI*, volume 9, pages 1420–1426, 2009.

[43] Andreas Kaltenbrunner, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael Banchs. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466, 2010.

[44] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 33. ACM, 2015.

[45] Hans Martin Espegren, Johannes Kristianslund, Henrik Andersson, and Kjetil Fagerholt. The static bicycle repositioning problem-literature survey and new formulation. In *International Conference on Computational Logistics*, pages 337–351. Springer, 2016.

[46] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[47] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222. SIAM, 2010.

[48] Richard J Oentaryo, Ee-Peng Lim, Jia-Wei Low, David Lo, and Michael Finegold. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 123–132. ACM, 2014.

[49] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[50] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

[51] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2006.

[52] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the national academy of sciences*, 101(11):3747–3752, 2004.

[53] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[54] Linton C Freeman, Stephen P Borgatti, and Douglas R White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks*, 13(2): 141–154, 1991.

[55] Mark EJ Newman. Analysis of weighted networks. *Physical review E*, 70(5):056131, 2004.

[56] Tore Opsahl, Vittoria Colizza, Pietro Panzarasa, and José J Ramasco. Prominence and control: the weighted rich-club effect. *Physical review letters*, 101(16):168702, 2008.

[57] Sahar Sohangir and Dingding Wang. Improved sqrt-cosine similarity measurement. *Journal of Big Data*, 4(1):25, 2017.

[58] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398, 2007.

[59] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[60] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[61] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[62] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov): 933–969, 2003.

[63] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96, 2005.

[64] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[65] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2013.

[66] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[67] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[68] Dheeraj kumar Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Role of matrix factorization model in collaborative filtering algorithm: A survey. *CoRR, abs/1503.07475*, 2015.

[69] Gregory T Knofczynski and Daniel Mundfrom. Sample sizes when using multiple linear regression for prediction. *Educational and psychological measurement*, 68(3): 431–442, 2008.

[70] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3 (2):1–159, 2012.

[71] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, 2013.

[72] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014.

[73] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.

[74] Zhengdong Lu, Berkant Savas, Wei Tang, and Inderjit S Dhillon. Supervised link prediction using multiple sources. In *2010 IEEE international conference on data mining*, pages 923–928. IEEE, 2010.

[75] Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, Berkant Savas, and Inderjit Dhillon. Scalable affiliation recommendation using auxiliary networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):1–20, 2011.

[76] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[77] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[78] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[79] V Dang. The lemur project-wiki-ranklib. *Lemur Project*, 2013.

[80] Dakshak Keerthi Chandra, Pengyang Wang, Jennifer Leopold, and Yanjie Fu. Collective embedding with feature importance: A unified approach for spatiotemporal network embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 615–624, 2020.

[81] Dakshak Keerthi Chandra, Kunpeng Liu, and Yanjie Fu. How unbalanced are bicycle dynamics? demand-supply shortage detection with spatiotemporal tensor factorization in station-less bike systems. In *SDM*, volume 18, page 18th, 2018.

[82] Dakshak Keerthi Chandra, Pengyang Wang, Jennifer Leopold, and Yanjie Fu. Collective representation learning on spatiotemporal heterogeneous information networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 319–328, 2019.

VITA

Dakshak Keerthi Chandra was born in Hassan city, Karnataka, India. He received his master's from Visvesvaraya Technological University, India. He began his Ph.D. study in January 2017 in the Department of Computer Science at Missouri University of Science and Technology. Under the advisement of Dr. Jennifer Leoplod, he focused on tackling open problems in Learning Representations of Heterogeneous Spatiotemporal Networks, and investigated strategies arising from graph theory, distributed machine learning, and Natural Language Processing. His studies as a Ph.D. student brought him to present in international conferences such as IEEE International Conference on Collaboration and Internet Computing (CIC), ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM International Conference on Information & Knowledge Management (CIKM), and IEEE International Conference on Big Data. Beyond his research, he interned at Dassault Systèmes as a Data Science and Machine Learning Developer. In May 2022, he received his Ph.D. degree in Computer Science from Missouri University of Science and Technology.