LEARNING SEMANTIC RELATIONSHIPS OF GEOGRAPHICAL AREAS BASED ON TRAJECTORIES

SAIM MEHMOOD

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO
MARCH 2020

Abstract

Mining trajectory data to find interesting patterns is of increasing research interest due to a broad range of useful applications, including analysis of transportation systems, location-based social networks, and crowd behavior. The primary focus of this research is to leverage the abundance of trajectory data to automatically and accurately learn *latent semantic relationships* between different geographical areas (e.g., semantically correlated neighborhoods of a city) as revealed by patterns of moving objects over time. While previous studies have utilized trajectories for this type of analysis at the level of a single geographical area, the results cannot be easily generalized to inform *comparative analysis* of different geographical areas. In this work, we study this problem systematically. First, we present a method that utilizes trajectories to learn low-dimensional representations of geographical areas in an embedded space. Then, we develop a statistical method that allows to quantify the degree to which real trajectories deviate from a theoretical *null model*. The method allows to (a) distinguish *geographical proximity* to *semantic proximity*, and (b) inform a comparative analysis of two (or more) models obtained by trajectories defined on different geographical areas. This deep analysis can improve readers understanding of how space is perceived by individuals and inform better decisions of urban planning. Our experimental evaluation aims to demonstrate the effectiveness and usefulness of the proposed statistical method in two large-scale real-world data sets coming from the New York City and the city of Porto, Portugal, respectively. The methods we present are generic and can be utilized to inform a number of useful applications, ranging from location-based services, such as point-of-interest recommendations, to finding semantic relationships between different cities.

Acknowledgements

I truly feel a sense of humbleness and gratitude towards my supervisor Prof Manos Papagelis for his support and guidance throughout the journey of my graduate studies. I am earnestly grateful for my lab mates Tilemachos Pechlivanoglou, Wenxiao Fu, Niloy Costa, Farzaneh Heidari et al. for being there at times of stress and anxiety. Also, I would like to thank the members of my dissertation committee: Dr. Zhen Ming (Jack) Jiang, Dr. Mojgan A. Jadidi and Dr. Hamzeh Khazaei for generously offering their time and guidance in terms of the improvement to this thesis document.

Table of Contents

Abstract **ii**

Acknowledgements **iii**

Table of Contents **iv**

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Research Questions & Contributions . . . 2

1.2 Thesis Organization . . . 3

2 Related Work **4**

2.1 Trajectory Data Mining . . . 4

2.2 Location-based Recommendations . . . 4

2.3 Trajectory-based Graph Embeddings . . . 5

2.4 Spatial Databases . . . 6

2.5 Expert Finding Techniques . . . 6

3 Learning Semantic Relationships of Geographical Areas **8**

3.1 Construction of a Uniform Grid . . . 9

3.2 Construction of a Lattice Graph 13

3.2.1 From Trajectories to Random Walks on a Lattice Graph 13

3.2.2 Learning Embeddings of Geographical Areas 14

3.3 Extensions 16

3.3.1 Points-of-Interest (POIs) & Cell Association 16

3.3.2 Translation of Trajectories into POIs 16

3.3.3 Learning Embeddings of POIs 19

4 Statistical Method for Distinguishing Geographical Proximity to Semantic Proximity 21

4.1 Models 21

4.1.1 Real Model 22

4.1.2 Null Model 22

4.1.3 Intermediate Model (An Alternate Null Model) 23

4.2 Model Analysis 23

4.2.1 Quantitative Analysis of Models 23

4.2.2 Exploratory Analysis of Models 27

5 Case Studies 28

5.1 Case Study I: New York City (NYC) 28

5.1.1 Data 28

5.1.2 Models 30

5.1.3 Exploratory Analysis of Models 30

5.1.4 Quantitative Analysis of Models 31

5.2 Case Study II: City of Porto 32

5.2.1 Data 32

5.2.2 Models 34

5.2.3 Exploratory Analysis of Models 34

5.2.4 Quantitative Analysis of Models . . . 35

6 System Implementation **44**

6.1 Spatial Database . . . 44

6.1.1 Modeling of Datasets in Database . . . 44

6.1.2 Spatial Relationships . . . 46

6.2 Environment . . . 49

7 Conclusion **50**

7.1 Future Work . . . 51

Bibliography **52**

List of Tables

3.1 Summary of Notations 12

5.1 Dataset Features NYC 30

5.2 Dataset Features City of Porto 34

List of Figures

3.1 Illustration of the proposed method for learning semantic relationships of geographical areas: (a) sample trajectories of the New York City taxi dataset, (b) trajectories traversing grid cells of a uniform grid, (c) trajectories as random walks on a lattice graph, (d) node embeddings of the lattice graph. 9

3.2 Sample grid and sample trajectory over grid cells. 11

3.3 Generating m walks from a single walk by using the process of trajectory permutations and feeding them to skip-gram. 15

3.4 Trajectories (red) representing object movements traversing through various grid cells (green) & relevant pois (blue) found inside grid cell nodes $n_{c_{ij}}$. 17

3.5 Trajectory (red) passing through grid cells c_{ij} (green) showing interaction with points-of-interest (blue). 19

3.6 Representing POIs embeddings based on trajectories movement across observational area A. . 20

4.1 Two grid cells (green rectangles) with several trajectories traversing both of them (shown as blue line), compared to trajectories that traverse either of them (shown as red line). 25

5.1 Sample trajectories of the New York City taxi dataset. 29

5.7 Sample trajectories of City of Porto taxi dataset. 33

5.2 Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models). 36

5.3 Exploratory Analysis of Models. (a), (b) and (c) provide the pair-wise similarities of a specific node $u = 23$ to all other nodes in the three models using heat maps (it allows for a one-to-many comparisons of models). 37

5.4 Cosine similarity between pairs of nodes. 38

5.5 Cosine similarity differences between pair of nodes. 38

5.6 Distribution of cosine similarity values between pair of nodes. 39

5.8 Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models). 40

5.9 Exploratory Analysis of Models. (a), (b) and (c) provide the pair-wise similarities of a specific node $u = 189$ to all other nodes in the three models using heat maps (it allows for a one-to-many comparisons of models). 41

5.10 Cosine similarity between pairs of nodes. 42

5.11 Cosine similarity differences between pair of nodes. 42

5.12 Distribution of cosine similarity values between pair of nodes. 43

6.1 Geometry primitives (2D) - PostGIS 45

1 Introduction

Advances in location acquisition and tracking devices have given rise to the generation of enormous trajectory data consisting of spatial and temporal information of moving objects, such as persons, vehicles or animals. These trajectories can either be physically constrained (e.g., a pedestrian walking on a sidewalk, vehicles driving on road network) or unconstrained (e.g., a birds flight). Discovering patterns and extracting knowledge from trajectories is critically important to many real-world applications, including human mobility understanding (e.g., pedestrian mobility mining), health care (e.g., detecting changes in gait patterns of seniors), smart transportation and urban planning (e.g., traffic forecasting and optimization), location-based services (e.g., recommendations of points of interest), to name a few. Harnessing the abundance of trajectory data and being able to design accurate predictive models can inform decision-making, and can enable cities to improve their operational efficiency and help their citizens to improve everyday living. Of great research interest have been problems related to trajectory similarity [34,35], trajectory clustering and outlier detection [18,42], or crowd behavioral analysis [6,10,29,32,43]. A comprehensive survey of classical trajectory data mining can be found in [44]. Recent advances on trajectory data mining look on *network dynamics of trajectories*, such as mining group patterns of trajectories [28,30] and mining the importance of a moving object in trajectory networks [26].

More recently, there is an increasing interest on utilizing geospatial information coming from trajectories to improve location-based recommendations using deep neural networks. The main idea of these approaches is to learn representations (embeddings) of points-of-interest (POIs) together with user profiles at the same low-rank space and then use the obtained embeddings to inform downstream data mining tasks [5,21,39].

Towards that end, different types of user trajectory profile properties have been exploited, such as *social influence or homophily* - users tend to follow their social network friends; *geographical proximity* - users tend to visit locations that are close to each other, around home or work; *periodicity* - users tend to visit same places at specific time intervals.

The primary focus of this research is to leverage the trajectory data to automatically and accurately learn latent *semantic relationships* between different geographical areas (e.g., semantically correlated neighborhoods of a city) as revealed by patterns of moving object trajectories over time.

1.1 Research Questions & Contributions

While previous studies have utilized trajectories for a similar type of analysis at the level of a single geographical area (e.g., a city), the results cannot be easily generalized to inform *comparative analysis of different geographical areas*. How people perceive different areas/neighborhood of their city? To what extend people in a city rely on geographical proximity of areas? Is the behavior of people of different geographical areas (e.g., two different cities) the same? If not, to what extend the behaviors are different? These are some of the motivating questions that we strive to answer in this research. we study these questions systematically and make the following major contributions:

- We present a method that utilizes trajectories to learn low-dimensional representations of the geographical areas that the trajectories span in an embedded space. The method relies on random-walk based methods for learning node representation of a graph and is able to reveal latent relationships of geographical areas, effectively defining semantic relationships between them. These latent semantic relationships can improve our understanding of how space is perceived by individuals (through their trajectories) and inform better decisions of urban planning.

- We develop a statistical method that allows to quantify the degree to which real trajectories deviate from a theoretical null model in a geographical area. The method allows to (a) distinguish geographical proximity to semantic proximity, (b) measure the extent of that difference (if any). Since the method is based on embedding trajectories on the same low-rank space, it allows to inform a comparative analysis between patterns of two (or more) models obtained by trajectories defined on different geographical areas (e.g., compare patterns in two different cities).

- We demonstrate the effectiveness and usefulness of the proposed embedding and statistical method in two case studies utilizing real-world data coming from the New York City and the city of Porto, Portugal, respectively.
- Lastly, we discuss the consequences of our work and how it can be extended to cover analysis on a different level of granularity, for example to include semantic analysis of points-of-interest (POIs).

The main contributions of the thesis have been published in the Proceedings of the 21st IEEE International Conference on Mobile Data Management 2020 [23].

1.2 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 reviews the existing work related to our research. Chapter 3 presents our method for learning semantic relationships of geographical areas and also extension discussing semantic analysis of points-of-interest. Chapter 4 formally presents the statistical model. In Chapter 5, we present two real-world case studies to demonstrate how the model can be applied in practice. After discussing the framework in Chapter 6, we conclude in Chapter 7.

2 Related Work

A number of important works related to our research has already been cited in introduction. Here we further elaborate on other related work.

2.1 Trajectory Data Mining

In this survey paper [44], the authors have explored the connections, correlations, and differences among trajectory pattern mining, trajectory data preprocessing, outlier detection and trajectory classification techniques. In our approach we are looking at trajectory pattern mining using unsuprevised learning methods, based on network representation. In [37] authors present a method of profiling moving objects by looking at their regional typical moving styles, which reflects geoinformation of the observed area and the moving behavior of objects. Shang et al. [31] are providing parallel collaborative method for trajectory to location join by addressing the challenges of spatiotemporal correlation between trajectories and locations and pruning the search space effectively. Similarly in [8], authors discuss how to enrich trajectories with semantic information based on stop points in moving objects. Kumar et al. [16] proposed a model that learns dynamic trajectory embeddings of users and items from a sequence of temporal interactions. We look at trajectories and geographical areas in terms of developing and enriching semantic understanding of regions.

2.2 Location-based Recommendations

Urban planning, relieving traffic congestion, and effective location recommendations are important objectives worldwide and have received increasing attention in recent years. In this survey [38], the authors are introducing methods used for location prediction and recommendations and giving an insight into trajectory data pre-processing for different objectives. In [3] authors are introducing realistic and financial aspects into trajectory data mining for bike sharing. They are designing a flexible objective function to tune the benefits between coverage of the number of users and the length of their trajectories. We are introducing computational aspects into our work by minimizing the cost of relating trajectories with geographical areas.

Recently, with the ease of access to acquire user activity records from large scale location-based social networks (LBSNs), many recent work has tried to improve location-based recommendation by exploiting various side effects of object movements [21]. Toblers first law [33] of geography is also an interesting concept to consider while thinking about semantic correlation between geographical regions i.e., "Everything is related to everything else, but near things are more related than distant things". Hao Wang et al. [36] proposed a latent probabilistic generative model called LSARS to mimic the decision-making process of users check-in activities both in home-town and out-of-town scenarios by adapting to user interest drift and crowd sentiments, which can learn location-aware and sentiment-aware individual interests from the contents of spatial items and user reviews.

2.3 Trajectory-based Graph Embeddings

Understanding user movement behavior is also among the challenges of location-based social networks. Cho et al. [4] has developed a model that captures human mobility based on periodicity and social ties. In [11] researchers have used student check-in data based on WiFi log files and proposed a network-based embedding method called *embedding for dense heterogeneous graphs*. Christoforidis et al. [5] adds an addition into the state-of-the-art work on using graph embeddings for points-of-interest recommendations by considering more spatial attributes around user generated spatiotemporal data-sets. Their work is similar to [5] as they are embedding different factors of user behavior in the same latent space. In [19] they are defining three types of different friends i.e., social, location and neighboring friends around user check-ins data. Their goal is also limited to standard recommendation, new users and new locations recommendations etc. In [48] authors have developed a framework towards learning trajectory context by adapting the problem to an encoder-decoder framework.

The idea of nodes and edges between nodes is fundamental to graph embedding approaches. We define nodes as small geographical regions and create edges between them based on adjacency. Further we take walks on these nodes by looking at trajectory paths through them. In [9], they propose an algorithmic framework for learning continuous feature representations for nodes in networks. Their key contribution is in defining a flexible notion of a nodes network neighborhood by choosing an appropriate notion using random walks. Similarly, we treat trajectory paths as walks and embed them in a low dimensional latent space.

2.4 Spatial Databases

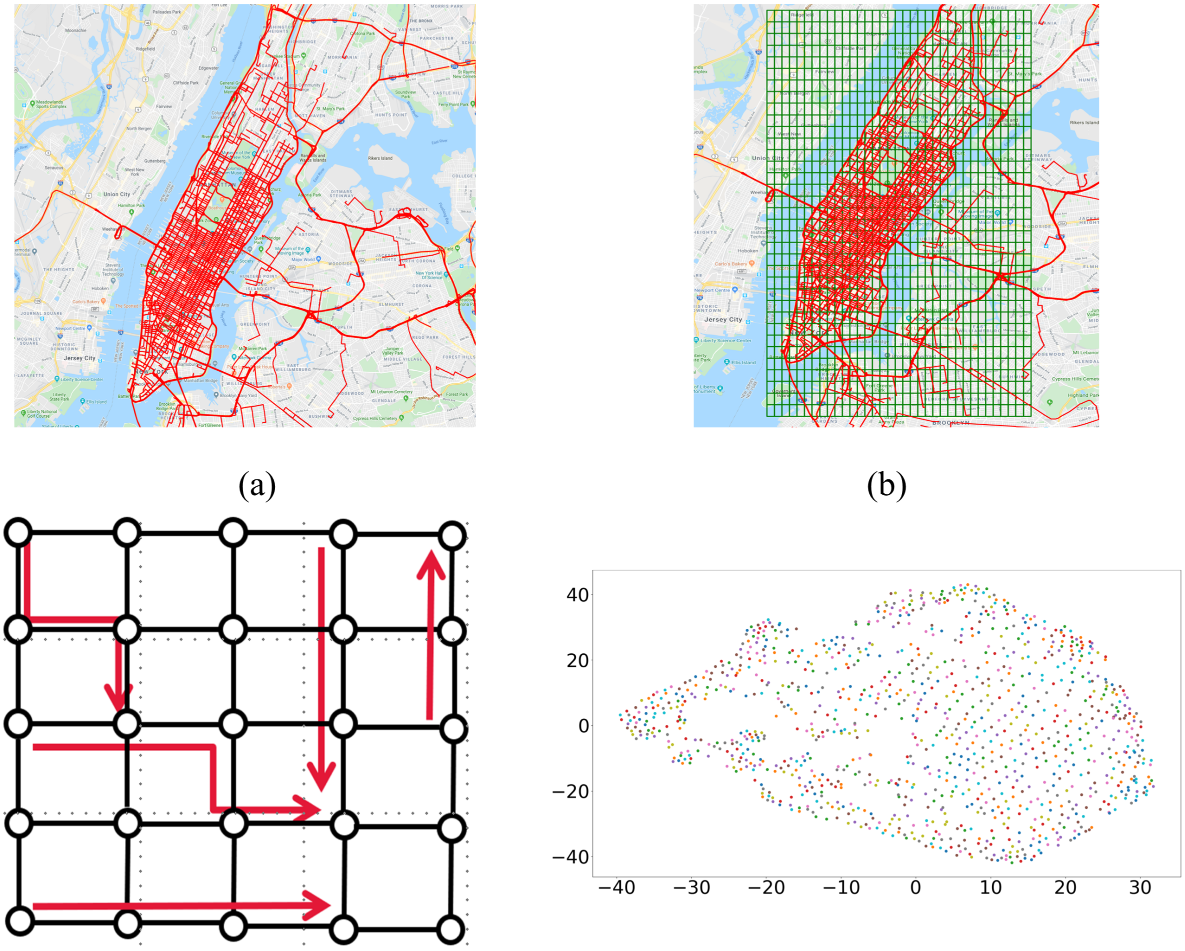
Ahmet et al. [15] introduced a data model PG-TRAJECTORY that is built on PostGIS, the spatial database extender of PostgreSQL. In their work they have introduced wide range of functions for storing and manipulating spatiotemporal trajectories. In [45–47] authors proposed methods to understand trajectories, mobility behavior of users and interesting locations by utilizing data-sets of users collected in Geolife project. Another work [40] utilizes trajectories for traffic state estimation. They have developed a framework that uses deep neural network to predict traffic states of each road individually from historical traffic information, along with prediction uncertainty. Further they refine these predictions by an ietartive boosting calibration procedure with embedded trajectories. In our work we are utilizing PostGIS and PostgreSQL to perform spatial queries on trajectory data-sets and trajectories and mobility behaviors are derived from the movement paths of objects.

2.5 Expert Finding Techniques

Finding experts in specified areas is an important task and has attracted much attention in the information retrieval community. Experts refer to people who are knowledgeable or who master in-depth skills in specified areas. In this survey paper [20] the authors talk about various techniques related to expert profiling such as *expert resource selection* - which extracts the expertise related data and information from which people with professional knowledge and skills can be discovered and *expertise modeling* - which builds expertise models to identify an expert. In [17] researchers are trying to find experts that have social connections and a set of required skills so that a team can be formed. Zhou et al. [41] have utilized social networking sites such Quora and Twitter to infer user expertise based on their tweets. They also consider relatedness between expertise topics as an important aspect in the inference process. Understanding geographical space including object interaction with points-of-interest leads to an interesting discovery and knowledge that can help us in developing expert profiles. These profiles can be built automatically by identifying and linking trajectories to moving objects and further assigning them expertise based on their interaction with points-of-interest and geographical space they are traversing.

3 Learning Semantic Relationships of Geographical Areas

In this section we describe a method that given a set of trajectories $T = \{t_0, t_1, t_2..., t_i..., t_{n-1}\}$, where t_i denotes the trajectory i defined over an observation area A, can learn semantic relationships of geographical areas of A. In brief, the method involves the following steps: (a) construction of a *uniform grid* that divides A to a set of evenly-spaced set of rows and columns (grid cells), (b) construction of a lattice graph based on the grid cells, (c) translation of trajectories as random walks on the lattice graph, and (d) use of (a variation of) a continuous skip-gram architecture model to learn distributed representations of nodes of the lattice graph, which effectively provide semantic relationships between geographical areas of A. Fig 3.1 shows the steps of the proposed method and Table I summarizes important notation.


(a)

(b)

(c)

(d)

Figure 3.1: Illustration of the proposed method for learning semantic relationships of geographical areas: (a) sample trajectories of the New York City taxi dataset, (b) trajectories traversing grid cells of a uniform grid, (c) trajectories as random walks on a lattice graph, (d) node embeddings of the lattice graph.

3.1 Construction of a Uniform Grid

Let a trajectory $t = \{(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\}$ amount to a route traveled from a starting point to an ending point, where the ordered sequence of pairs (x, y) represent latitude and longitude coordinates in the

2D Cartesian system. We can also represent trajectories as an ordered sequence of points:

$$t = \{p_0, p_1, p_2..., p_n\} \tag{3.1}$$

where p_{i-1} is the *ith* point of the trajectory t. While individual trajectories of moving objects are defined at a lower level of granularity (e.g., sequences of pairs of longitude and latitude coordinates), analysis of geographical areas typically needs to be done at a higher level of granularity, such as the level of neighborhoods or postal codes of a city. Without loss of generality, we adopt the abstraction of a *uniform grid* that divides the observation space A to a set of evenly-spaced set of r rows and c columns, forming *grid cells*; row height does not need to be equal to column width. Formally, we define a $grid_{rc}$ as follows:

$$grid_{rc} = \{c_{00}, c_{01}, c_{10}, ..., c_{r-1c-1}\} \tag{3.2}$$

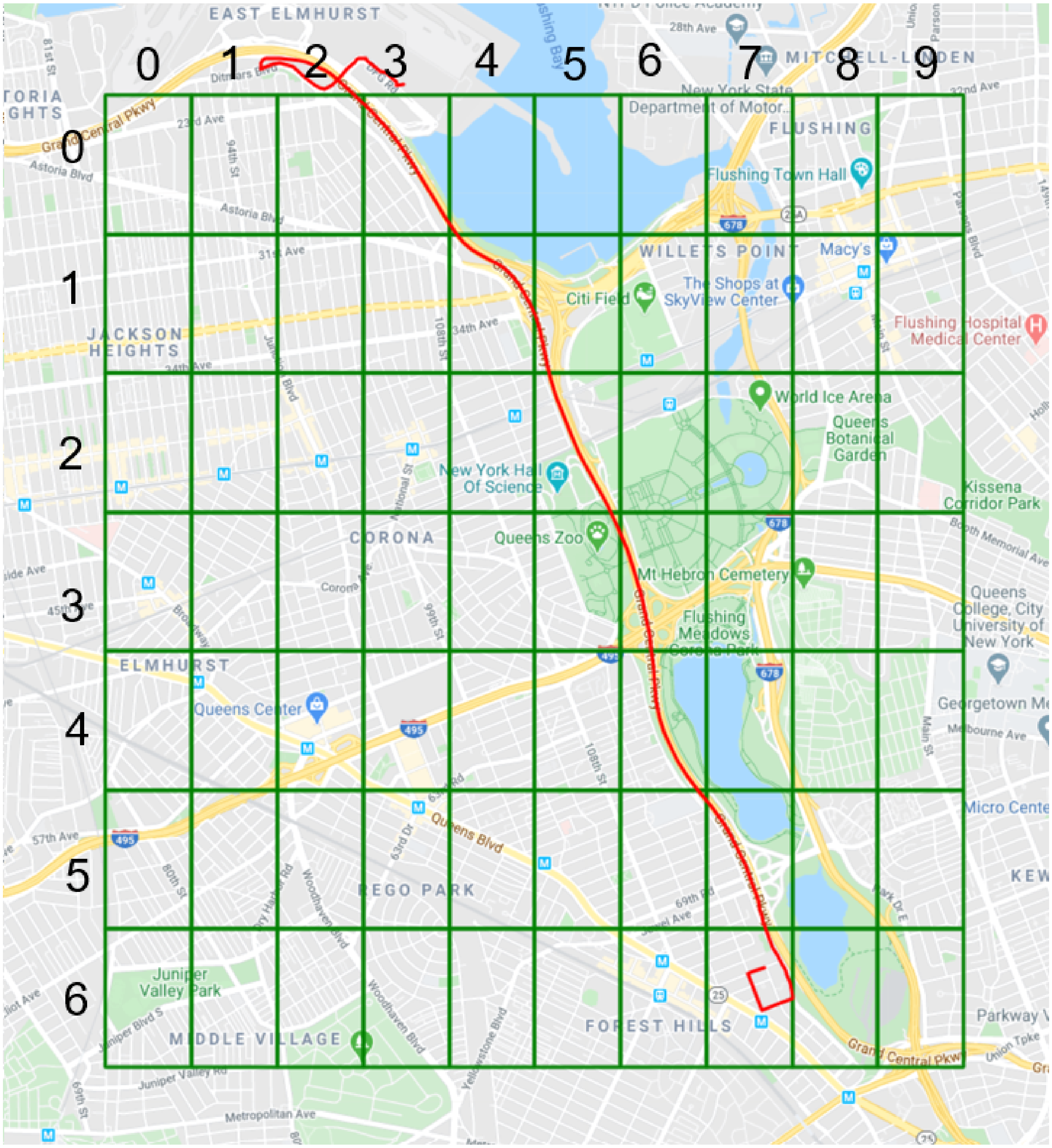


Figure 3.2: Sample grid and sample trajectory over grid cells.

The $grid_{rc}$ consists of $r \times c$ grid cells and c_{ij} is representing the grid cell at row i and column j. By dividing A into grid cells we are able to translate a trajectory $t \in T$ from a sequence of geolocations to

Table 3.1: Summary of Notations

Symbol	**Description**
T	set of trajectories $T = \{t_0, t_1, t_2 ..., t_i ..., t_n\}$
P	set of points-of-interest $P = \{poi_1, poi_2, ..., poi_i, ..., poi_n\}$
A	observation space
$grid_{rc}$	a uniform grid of r rows and c columns $grid_{rc} = \{c_{00}, c_{01}, c_{10}, ..., c_{ij}, ..., c_{rc}\}$
$n_{c_{ij}}$	representing all grid cells c_{ij}
c_{ij}	the grid cell at the ith row and jth column
G	lattice or grid graph
$\mathbb{R}^n$	low dimensional latent space
v_i	vertex representing grid cells c_{ij}
$e_{(u,v)}$	edge between adjacent nodes
w	windows-size $w = 10$
λ_a	threshold for cosine similarity
λ_b	threshold for cosine similarity difference
W_{v_i}	Random walk on grid cells

a sequence of grid cells on $grid_{rc}$. For example, Fig. 3.2 shows a $7x10$ grid and a sample trajectory that traverses 12 grid cells, starting at c_{02} moving to c_{03}, c_{04}, c_{14}, c_{15}, ..., all the way to c_{67}. Note that the size of the grid provides an interesting trade-off between a more refined analysis and a faster analysis. This is because the larger the number of rows and columns of the grid, the smaller the geographical areas represented by each grid cell, but at the cost of having to associate each trajectory to a larger number of grid cells, which is computationally more expensive.

3.2 Construction of a Lattice Graph

Given a $grid_{rc}$ we can construct a lattice graph $G(V, E)$ of V nodes and E edges, where any node $n_{c_{ij}} \in V$ of the lattice represents a grid cell $c_{ij} \in grid_{rc}$ and an edge $e_{(u,v)} \in E$ represents that grid cells u and v are adjacent in the $grid_{rc}$. A lattice or grid graph, is a graph whose drawing, embedded in some Euclidean space forms a regular tiling. As a trajectory traverses tiles of a grid, this traversal can also be modeled as a walk on the lattice graph G.

3.2.1 From Trajectories to Random Walks on a Lattice Graph

We briefly entertained the idea of treating real-world trajectories as walks on a lattice graph. The motivation is that random walks on a graph have been successfully used as a way to obtain semantic relationships between nodes of a graph [9,27]. Therefore, we use this analogy to learn relationships between different geographical regions that could be far apart in Euclidean space. Intuitively, the main hypothesis is that nodes that are found multiple times in a large number of different random walks, they probably share some semantic similarity and should be embedded closer together, even though they might not be close to each other. We intent to exploit this **key idea** to learn semantic relationships between geographical areas that can be far apart in Euclidean space.

Formally, random walks are denoted as W_{v_i}, where v_i denotes a vertex. They represent a stochastic process with random variables $W_{v_i}^1$, $W_{v_i}^2$, ... , $W_{v_i}^k$ such that $W_{v_i}^{k+1}$ is a vertex chosen at random from the neighbors of vertex v_k. Random walks have been used for variety of problems such as content recommendation and community detection [2] [7], in different kinds of networks. Nodes in a network can be classified on the basis of *homophily* and *structural equivalence* [14] roles. According to homophily hypothesis, nodes that are close by and belong to similar network communities should be embedded closer to each other. For example, in Fig. 3.2, the grid cells c_{00}, c_{01} and c_{10} represent same local network community, as they are connected to each other. On the other hand, structural equivalence describe nodes that have similar structural roles in networks and should be embedded close to each other. For example, in Fig 3.2, the grid cells c_{02}, c_{03}, ..., c_{61}

that a trajectory traverses have the same structural role, as they are all part of the same trajectory.

3.2.2 Learning Embeddings of Geographical Areas

We describe how given a graph and a set of random walks defined over its nodes, we can obtain node embeddings that will bring similar nodes closer to each other in the embedded space. Given an undirected and unweighted graph $G = (V, E)$, we aim to learn the mapping function $f : V \rightarrow \mathbb{R}^d$, where d is the network representation dimension and each row is the vector representation of a node. The training objective function is to maximize the log-probability of the nodes appearing in the context of the node v_i. Context of each node v_i is provided by setting a window-size w that defines a set of nodes of the random walk W_{v_i} around v_i, similar to the process described in previous work [27]. Using that approximation objective and the skip-gram model of *node2vec* [9], we obtain embeddings that are optimized by stochastic gradient decent so that:

$$Pr(v_j|\mathbf{v_i}) \propto \exp\left(\mathbf{v_j^T}\mathbf{v_i}\right) \tag{3.3}$$

where $\mathbf{v_i}$ is the vector representation of a node v_i ($f(v_i) = \mathbf{v_i}$). $Pr(v_j|\mathbf{v_i})$ is the probability of the observation of neighbor node v_j, within the window-size given that the window contains v_i. In our experiments, we use the `gensim` implementation of the skip-gram model[1]. We set the window size to $w = 10$ and the number of dimensions to $d = 128$. A similar approach has been employed in [12, 13] to learn low-rank embeddings of evolving networks.

Trajectory Permutations: The `skip-gram` model described in the previous paragraph is based on the distributional hypothesis [24] that suggests that the more semantically similar two nodes are, the more they will tend to occur in similar contexts. As the use of the `skip-gram` originates in word embeddings, typically the context is defined by a small window size (e.g., $w = 5$ is common) that defines the surrounding words of a target word in a sentence. By design, the `skip-gram` architecture weighs nearby context words more heavily than more distant context words. However, in the case of trajectories, it is important that every node v_j in a walk W_{v_i} (i.e,. in a trajectory) appears in the context of every other node irrelevant of how

[1] https://github.com/RaRe-Technologies/gensim

far they are from each other. To achieve that, we rely on generating m random permutations of a single trajectory and providing these m trajectories to the `skip-gram` model (as shown in Fig 3.3). Recall that every trajectory $t \in T$ represents an ordered list of the grid cells traversed, and can be represented as a single walk on the lattice graph. Formally, let a single walk $W_{v_i} = \{v_i, ..., v_k\}$ starting from vertex v_i and ending at vertex v_k. The number of permutations on a set of k elements is given by $k!$, which can be a very large number. Instead, we create only a fixed number of $m \ll k!$ random permutations and use these as input walks to the model. To obtain m rearrangements of the elements of an ordered list, we generate m permutations for each walk, so that each vertex v_i in a walk has a chance to appear in different positions. As a result, we end up with m times more walks. This process will effectively neutralize the effect of each context node and distant nodes will not be less weighted than more nearby context nodes. Alternatively, one could employ the continuous bag-of-words (cbow) architecture that follows the bag-of-words assumption and treats surrounding context nodes equally (i.e., the order of context nodes does not influence prediction). However, in that case, the window-size defining the context would need to be adjusted every time to the length of a single trajectory, but the embedding model we rely on assumes a fixed window-size.

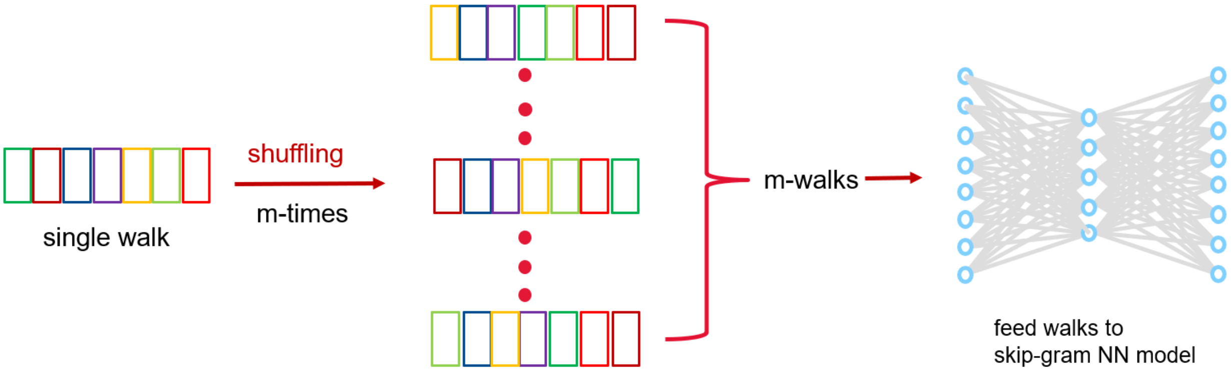


Figure 3.3: Generating m walks from a single walk by using the process of trajectory permutations and feeding them to skip-gram.

3.3 Extensions

This section discusses the consequences of our work and how it can be extended to cover analysis on a different level of granularity i.e., to include semantic analysis of points-of-interests.

3.3.1 Points-of-Interest (POIs) & Cell Association

We can associate POIs with grid cell c_{ij} based on the metric that all POIs which are inside the cell boundary coordinates belong to the cell. This gives a bird's-eye view of the geographical region and can help us in identifying places of interest with less computational cost.

Association between a cell and its POIs can be mathematically written as:

$$c_{ij} = \{poi_1, poi_2, ..., poi_i, ..., poi_n\} \tag{3.4}$$

where c_{ij} refers to a cell in ith row and jth column inside a dynamic grid $grid_{rc}$.

3.3.2 Translation of Trajectories into POIs

Trajectories inside a grid cell c_{ij} can be associated with POIs $poi_1, ...poi_n$ if they are inside the same c_{ij}.

Relating Trajectory t_i with POIs can be mathematically written as:

$$P^{t_i} \subseteq P = \{poi_1, poi_2, ..., poi_i, ..., poi_n\} \tag{3.5}$$

where P^{t_i} represents POIs that belongs to trajectory t_i.

Considering grid cells, trajectories and POIs concurrently in Fig 3.4, we can represent trajectories as points-of-interests i.e., such that we take the cells (e.g., in Fig 3.2, starting at c_{00} moving to c_{10}, c_{11}, c_{21}, c_{22}, ..., all the way to c_{61}) through which trajectory is passing and create walks W_{v_i} based on all the POIs inside those cells. The advantage comes in the form of ease of introducing *geographical proximity* and *preference dynamics* into user trajectory paths. By looking at a single grid cell, POIs inside its boundary and trajectory portion (as in eq. 3.1 $\{p_i...p_k\}$) passing through, it enables us to associate only those POIs which are inside the c_{ij} i.e., thus considering geographical proximity of user trajectory path. User preference dynamics is added by minimizing c_{ij} size which reduces the number of POIs inside cell. This generates an understanding regarding which POIs were visited by users in the order of sequence.

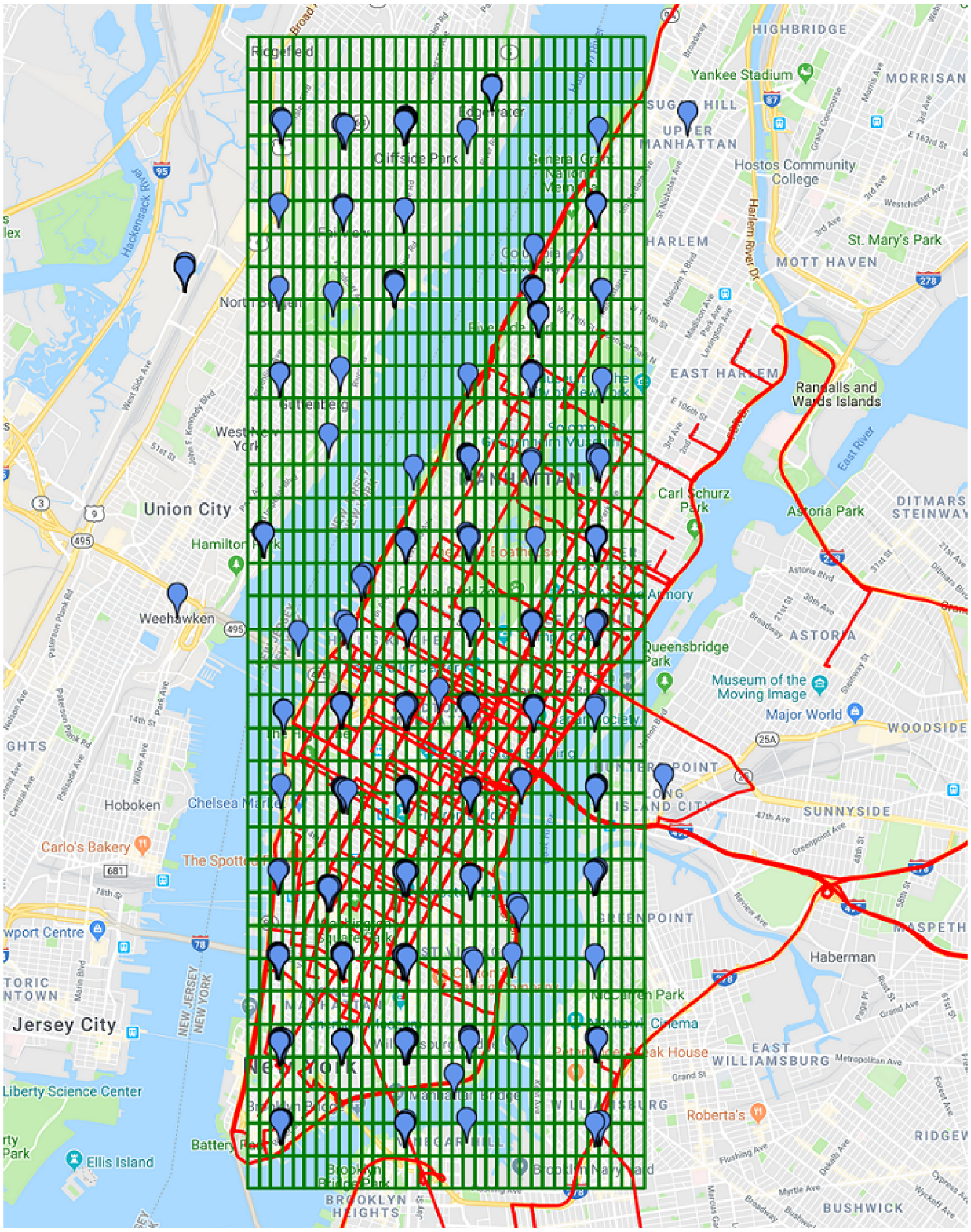


Figure 3.4: Trajectories (red) representing object movements traversing through various grid cells (green) & relevant pois (blue) found inside grid cell nodes $n_{c_{ij}}$.

Use Case: Business Recommendation Trajectory association with POIs $\{poi_1, poi_2, ..., poi_i, ..., poi_n\}$ based on c_{ij} has an interesting business use case that enables us to recommend businesses to relevant people. From Fig. 3.5 it can be observed that recommending POIs to the trajectory passing through relevant grid cells makes it more intuitive. As the relevant user would have more probability of interacting with these businesses.

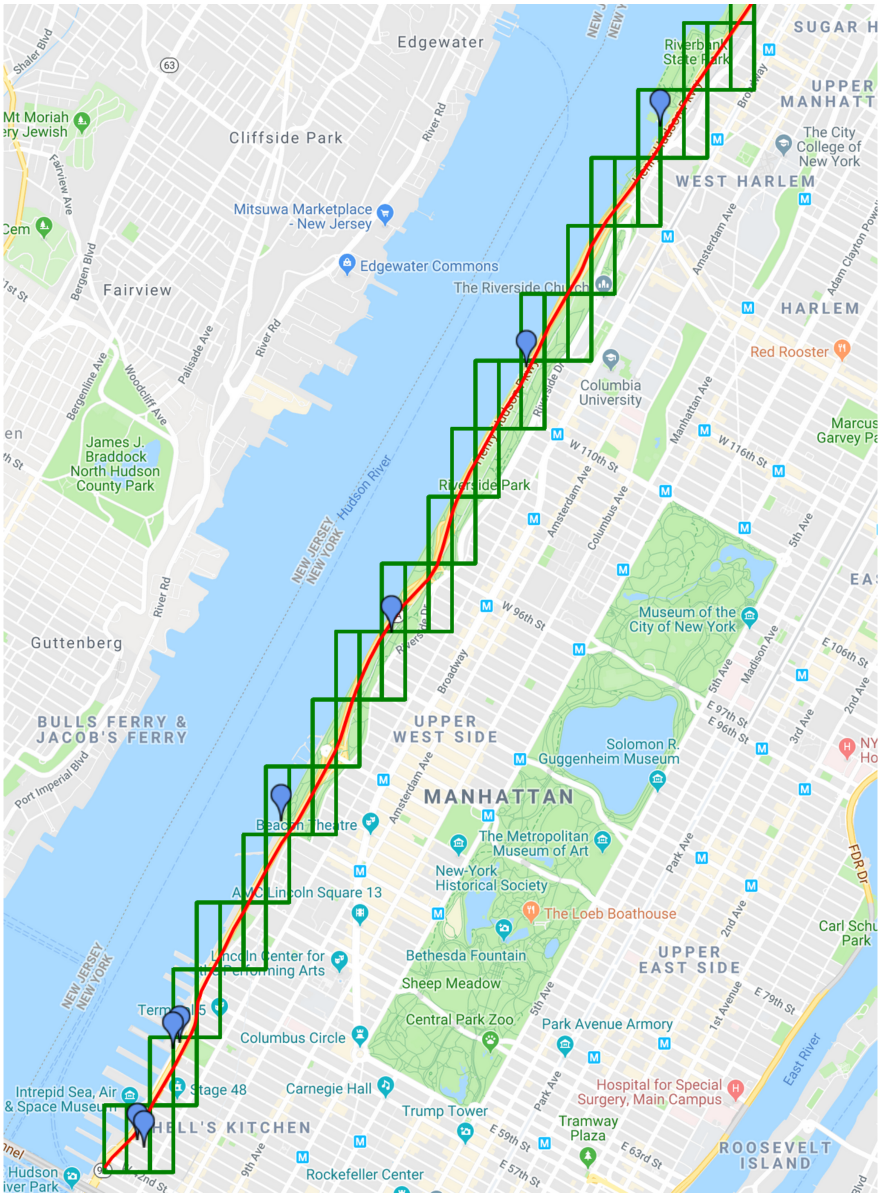


Figure 3.5: Trajectory (red) passing through grid cells c_{ij} (green) showing interaction with points-of-interest (blue).

3.3.3 Learning Embeddings of POIs

As discussed previously, by considering $grid_{rc}$, T & P concurrently we create walks of P^{t_i} i.e., each walk represents sequence of P which are in the close vicinity of T. Similar to the process described for *Learning*

Embeddings of Geographical Areas we learn embeddings of POIs.

Node embeddings based on trajectory movements shows structural equivalence as they are embedded closely based on their role for being connected to the same trajectory. Node embeddings based on random walks specifies homophily, as nodes are embedded closed to each other based on adjacency. Node embeddings based on trajectory representation as POIs represent P^{t_i} that are in the close proximity of the trajectory. This help us in identifying user interests and built expert profiles - which can be implemented as a future work. Embedding visualization of walks based on POIs is shown in Fig. 3.6.

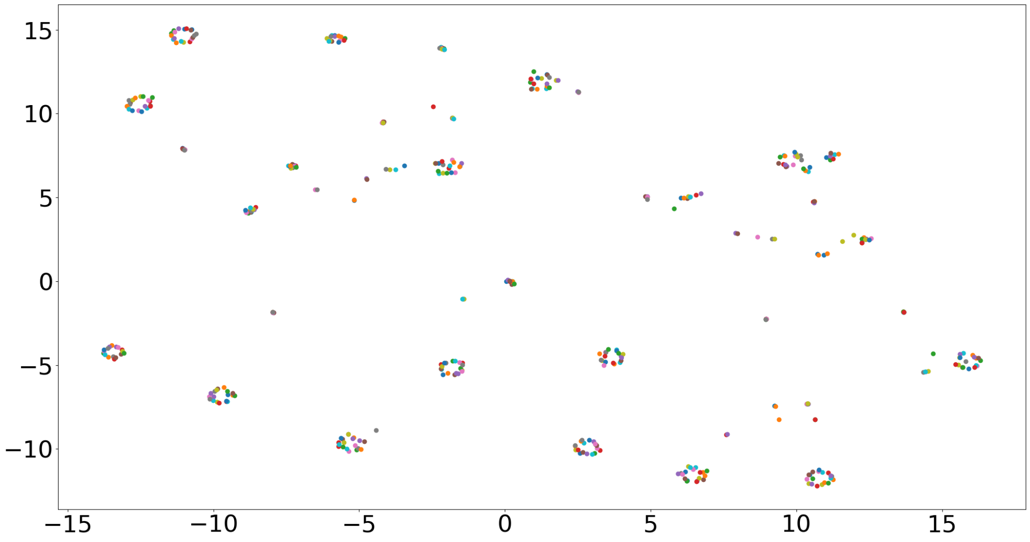


Figure 3.6: Representing POIs embeddings based on trajectories movement across observational area A.

Where clusters of POIs represents P^{t_i} belonging to similar trajectories T i.e., trajectories taking similar routes in the observational area A.

4 Statistical Method for Distinguishing Geographical Proximity to Semantic Proximity

Up till now, we have presented a method that given a large set of trajectories T defined over an observation area A (i.e., a city) can learn semantic relationships of geographical areas of A, in an unsupervised fashion. In this section, we present a statistical method for determining whether the observed data (i.e., the learned representations) display non-trivial properties that would not be expected on basis of chance alone. To that end, we design a *null model* that behaves in accordance with a reasonable null hypothesis for the behavior in question (i.e., how objects or people move in an area?). The null hypothesis is based on the assumption that people perceive a city based on geographical proximity, so that the chance to move from one place to another is dictated by physical separation. We provide details of the real model and the null model below. We also investigate an *alternate null model* that matches an additional feature of the observed model in question (the origin of the trajectory). We call that model, *intermediate model.* Then, we provide methods for quantitative and visual comparison of different models that can help compare the null model(s) to the real one and allow to inform conclusions. The idea of using a randomization technique to distinguish real observations from a theoretical null model has been successfully employed in various settings, such as in distinguishing influence from correlation in a social network [1, 25].

4.1 Models

Given a large set of trajectories T defined over an observation area A. we define three models for analysis: the *real model*, the *null model* and the *intermediate model*, which serves as an alternate null model. For all models, the same fixed size uniform grid $grid_{rc}$ of r rows and c columns is applied that leads to the construction of the same lattice graph $G(V, E)$.

4.1.1 Real Model

This model is generated by utilizing a subset $S \subseteq T$ of the set of real trajectories, such that $|S| \ll |T|$. Every sample trajectory $s \in S$ is selected uniformly at random from T. Given A, $grid_{rc}$ and S, we obtain vector representations for each geographical area of A, defined by $grid_{rc}$. The learned vectors will be used to analyze the semantic relationship between the geographical areas.

4.1.2 Null Model

This model is generated by defining random walks over the lattice graph. A walk starts at a node in the lattice graph and at every step moves to one of its adjacent nodes. The random walks are obtained by `node2vec` with default values, which suggests generating ten (10) random walks for each node. However, we constrain the random walk length of each walk to be equal to the average length of the walks defined by the trajectories in the sample S of the real model. This is important, as an appropriate null model needs to satisfy some constraints coming from the real model, but which is otherwise taken to be an unbiased random structure. So, we set the random walk length parameter of `node2vec` to be $walk - length = \sum_{i=1}^{|S|} \ell_{s_i} / |S|$, where ℓ_{s_i} represents the length of the s_i random walk in the lattice graph G (i.e, the number of grid cells of $grid_{rc}$ that the s_i trajectory has traversed). This parameter specifies how many other nodes will be visited by a walk. Given A, $grid_{rc}$ and S, we obtain vector representations for each geographical area of A, defined by $grid_{rc}$. The learned vectors of the null model effectively cover every node in the entire observation region A and choices of the random walks are dictated by geographic proximity. We will be comparing this theoretical null model to the real model.

4.1.3 Intermediate Model (An Alternate Null Model)

This model serves as an alternate null model. While in the null model we only constrained the random walk length of each walk to be equal to the average length of the walks defined by the trajectories in the sample S of the real model, in the intermediate model we consider *two additional constraints*: (i) the number of walks are equal to the number of trajectories in S, and (ii) the origin node from which a random walk starts is defined by the first node of each $s_i \in S$. The main motivation for this model is to learn vectors that can capture more of the constraints of the real trajectories, but still maintain the unbiased random walks on the lattice graph that are still dictated by geographic proximity. We will be comparing the intermediate model to the null and the real model.

4.2 Model Analysis

Each of the aforementioned models learns a low-dimensional vector representation for each graph node (i.e., for each geographical area of A). Here, we present metrics that allow to compare the models both *quantitatively* and *visually*. The former allows to test the the null hypothesis (accepting or rejecting it) and also to numerically compare the descriptive analytics of each model. The latter allow for exploratory data analysis, which helps to visually summarize the main characteristics of the models.

4.2.1 Quantitative Analysis of Models

The most significant metric of our analysis is the pair-wise similarity of nodes of the graph. This metric allows to find pairs of nodes that are related based on trajectory data patterns. This metric also allows to discover interesting pairs of nodes. These are pairs of nodes that expose a large difference (of their similarity score) in two different models, therefore shedding light in "unexpected" semantic relationships that cannot be explained by geographical proximity. To compare two models we also provide a metric of distance between two distributions of pair-wise similarity values using normalized histograms; a statistical test/metric is presented that can be used to determine whether there is a statistically significant difference (i.e., a magnitude of difference that is unlikely to be due to chance alone) between the real model and any of the null models.

Cosine Similarity Between Nodes A common method to calculate a similarity score between two vector embeddings is to use cosine similarity, which is a measure of similarity between two non-zero vectors that measures the cosine of the angle between them. Formally, given the vectors $\vec{\mathbf{v_i}}$ and $\vec{\mathbf{v_j}}$ of nodes i and j, their cosine similarity is given by:

$$cos\theta(\vec{\mathbf{v_i}}, \vec{\mathbf{v_j}}) = \frac{\vec{\mathbf{v_i}}.\vec{\mathbf{v_j}}}{\|\vec{\mathbf{v_i}}\|.\|\vec{\mathbf{v_j}}\|} = \frac{\sum_1^n \mathbf{v_i}\mathbf{v_j}}{\sqrt{\sum_1^n \mathbf{v_i}^2}\sqrt{\sum_1^n \mathbf{v_j}^2}} \tag{4.1}$$

where $\vec{\mathbf{v_i}}.\vec{\mathbf{v_j}} = \sum_1^n \mathbf{v_i}\mathbf{v_j} = \mathbf{v_{i1}}\mathbf{v_{j1}} + \mathbf{v_{i2}}\mathbf{v_{j2}} + ... + \mathbf{v_{in}}\mathbf{v_{jn}}$ is the dot product of the two vectors. Cosine similarity is particularly used in positive space, where the outcome is bounded in $[0, 1]$. We are adopting this interpretation and we ignore the pairs of nodes whose cosine similarity is negative. Note as well that the purpose of calculating pair-wise cosine similarities is to discover any semantic relationships between nodes, and a negative cosine similarity indicates that two nodes are not related to each other. Depending on domain expertise, one can define a threshold value λ_a, such that if the cosine similarity between a pair of nodes is equal to or greater than λ_a (i.e., $cos\theta \geq \lambda_a$), then the pair of nodes is considered "similar". Even if such a domain knowledge is not always available, we can still identify pairs of nodes that exhibit different similarity in different models. This brings us closer to our initial motivation, which is the ability to identify geographical areas in an observation area A that are semantically similar and this similarity cannot be attributed to chance. Take for example, Fig. 4.1 that shows an example of two grid cells (i.e., nodes) that while they are geographically far apart ($cos\theta = 0.41$ in the null model; $cos\theta = 0.45$ in the intermediate model), they are semantically similar in the real model ($cos\theta = 0.73$ in the real model). This can be attributed to the fact that there are many trajectories that traverse from both these grid cells (shown as blue lines), compared to trajectories that are traversing through either of the grid cells (shown as red lines).

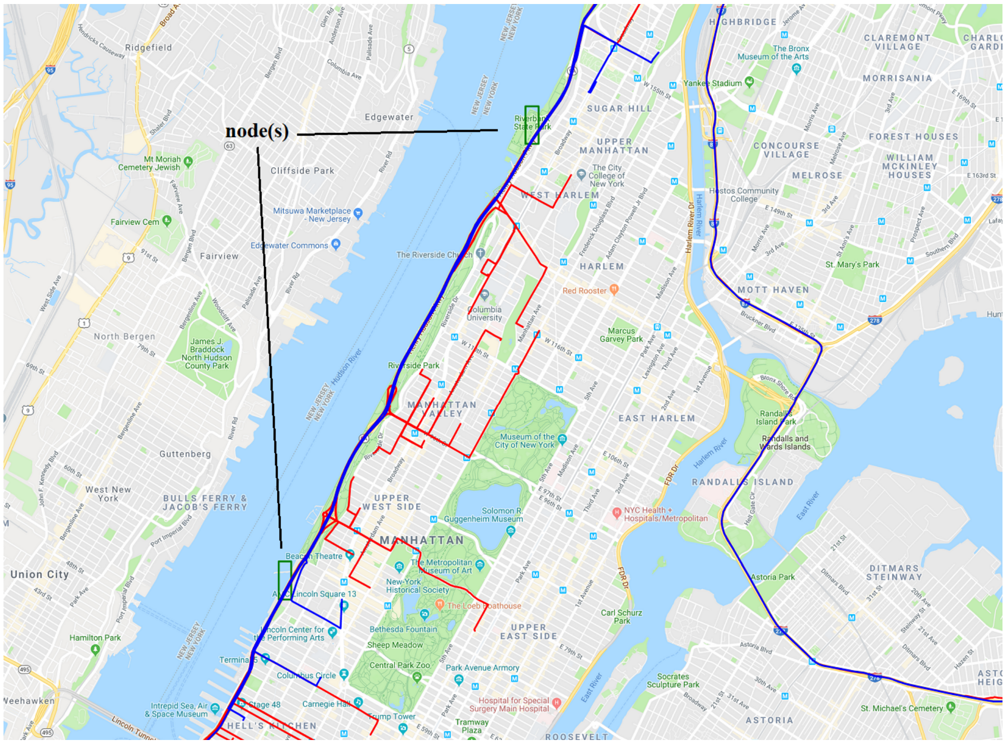


Figure 4.1: Two grid cells (green rectangles) with several trajectories traversing both of them (shown as blue line), compared to trajectories that traverse either of them (shown as red line).

Discovery of Interesting Pairs of Nodes By comparing the similarity of pairs of nodes in different models, it is possible to discover "interesting" ones. These are pairs of nodes that expose a large difference of their similarity score in two underlying models (e.g., real vs null model). Formally, given two models X and Y, it is:

$$d_{X,Y}(\vec{\mathbf{v_i}},\vec{\mathbf{v_j}}) = |cos\theta_X(\vec{\mathbf{v_i}},\vec{\mathbf{v_j}}) - cos\theta_Y(\vec{\mathbf{v_i}},\vec{\mathbf{v_j}})| \tag{4.2}$$

Depending on domain expertise, one can define a threshold value λ_b, such that if the cosine similarity difference of a pair of nodes in different models is equal to or greater than λ_b (i.e., $d_{X,Y}(\vec{\mathbf{v_i}},\vec{\mathbf{v_j}}) \geq \lambda_b$), then the pair of nodes is considered "interesting". Apparently, a pair of nodes is interesting when their associated vectors are found to be very similar in one model and dissimilar in the other, or the other way around. Intuitively, these pairs of nodes are characterized as interesting because they reflect geographical areas in the observation space A that are perceived by people living and travelling in A as being semantically similar (based on large trajectory data patterns). These similarities cannot be explained by geographical proximity, and therefore they cannot be attributed to chance (as depicted by the null model).

Distribution of Pair-wise Similarities We are interested in comparing the distribution of pair-wise similarity of nodes in different models. Towards that end, we construct a histogram for each model, where each bin represents a range of cosine similarity values and then count the number of pairs that belong to each bin. Effectively, a histogram allows to show the underlying frequency distribution of a set of continuous values (in our case the cosine similarity values between pairs of nodes). This allows to inspect the data for its underlying distribution and to use them to compare different models. Formally, for every model we construct a histogram as a function m_i that counts the number of observations (i.e., pair-wise similarity) that fall into each of the disjoint similarity categories (bins) – we define 100 equal bins in the range $[0, 1]$. Let n be the total number of observations and b be the total number of bins, then the histogram m_i is given by:

$$n = \sum_{i=1}^{b} m_i \tag{4.3}$$

A cumulative histogram is also possible that counts the cumulative number of observations in all bins up to a specified bin. The cumulative histogram M_i of a histogram m_j is given by:

$$M_i = \sum_{j=1}^{i} m_j \tag{4.4}$$

To compare two models, we rely on comparing the distance between two histograms H^A, H^B. There are many metrics for comparing the distance between two histograms, including a chi-square or KolmogorovSmirnov test statistic. For simplicity, we employ a chi-square distance:

$$\chi^2 = d(H^A, H^B) = \sum_{i=1}^{b} \frac{(H_i^A - H_i^B)^2}{H_i^A} \tag{4.5}$$

where b is the number of bins and H_i^A and H_i^B are the values of the ith bin in the histograms H^A and H^B, respectively.

4.2.2 Exploratory Analysis of Models

Exploratory data analysis helps to visually summarize the main characteristics of data. To that end, we develop metrics that allow to visually compare the models. In particular, we present a many-to-many visualization that can provide a summary of how embedded vectors are organized in low-dimensional space. We also provide a one-to-many visualization based on heat maps that can illustrate the similarity of a single predefined node u to all other nodes in the analysis.

Model Embeddings (Many-to-many Visualisation) T-distributed Stochastic Neighbor Embedding (t-SNE) [22] is a machine learning algorithm for embedding high-dimensional data for visualization in a low-dimensional space of two (or three) dimensions. We rely on 2D t-SNE to visualize the learned vector embeddings of the different models of the analysis. Specifically, the t-SNE visualization plots each 128-dimensional node as a two-dimensional point in such a way that similar nodes are shown nearby and dissimilar nodes are shown as distant points with high probability. As a result, we can obtain a visual summary of the main behavior of the models and quickly determine whether they are similar or not.

Heat map (One-to-many Visualisation) A heat map is a graphical representation of data that uses a system of color-coding to represent different values. Typically, larger values are represented by darker colors and smaller values are represented by lighter colors. We use heat map to show the vector similarity between *a predefined node* u and all other nodes in the analysis. When multiple heat maps of the same node u are shown for different models, then this visualization helps to identify nodes that might be semantically similar to u in one model, but not in the other model. Recall that in the real model, similarities are due to patterns of real trajectories, while in the null model they are due to geographical proximity.

5 Case Studies

To demonstrate the effectiveness and usefulness of the proposed embedding and statistical method, we design two large-scale case studies utilizing real-world data coming from the New York City and the city of Porto, Portugal, respectively.

5.1 Case Study I: New York City (NYC)

5.1.1 Data

This dataset is released by NYC Taxi and Limousine Commission (TLC), which includes pickup & dropoff time, geo-coordinates, number of passengers, and several other features. The data-set file contains $1,458,644$ trip records and features containing pickup and dropoff points as pairs of (longitude, latitudes) coordinates. For the needs of our study we rely on a random sample S that includes $10,000$ trajectories. For each pair of pickup and dropoff locations, we utilize the Google Directions API to create trajectories in NYC. Description of features is mentioned in table 5.1:

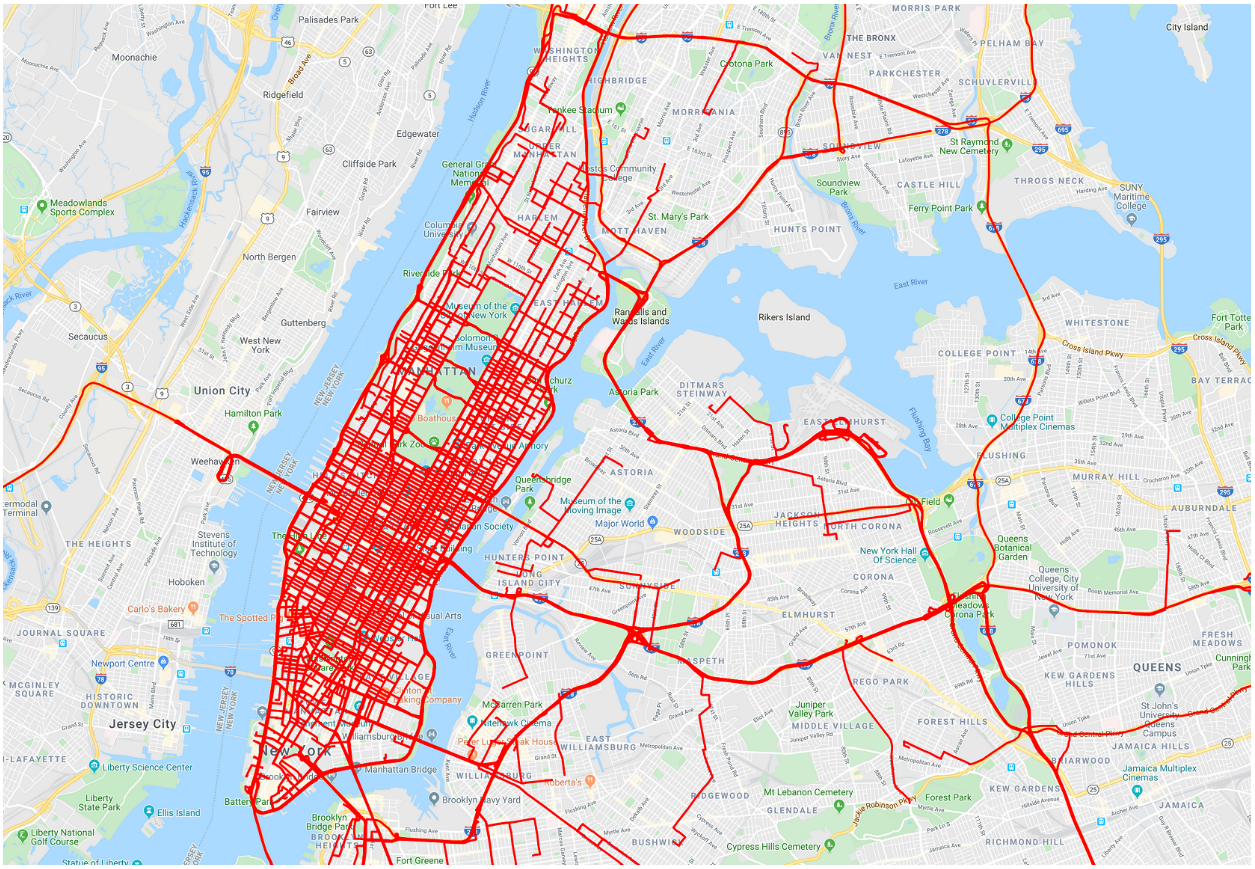


Figure 5.1: Sample trajectories of the New York City taxi dataset.

Table 5.1: Dataset Features NYC

Columns	**Description**
id	a unique identifier for each trip
vendor_id	a code indicating the provider associated with trip record
pickup_datetime	date and time when the meter was engaged
dropoff_datetime	date and time when the meter was disengaged
passenger_count	the number of passengers in the vehicle
pickup_longitude	the longitude where the meter was engaged
pickup_latitude	the latitude where the meter was engaged
dropoff_longitude	the longitude where the meter was disengaged
dropoff_latitude	the latitude where the meter was disengaged
trip_duration	duration of the trip in seconds

5.1.2 Models

Given an observation area A defined by trajectories in the NYC city, a set of trajectories S and a uniform grid $grid_{rc}$ with $r = 35$ rows and $c = 35$ columns, we use the methods described in Chapter 3 and 4 to obtain three models: the *real model*, the *null model* and the *intermediate model.*

5.1.3 Exploratory Analysis of Models

We begin the analysis of the models with an exploratory analysis.

First, we employ the t-SNE method to visually summarize the learned representations of the three models. Each point in the 2D visualization represents one of the $35 \times 35 = 1225$ uniform grid cells (i.e., geographical areas of A) for which a 128-dimensional representation has been learned based on the set of S sample trajectories. Fig. 5.2a, 5.2b, and 5.2c show the results for the three models, respectively. The visualization succeeds in revealing some significant differences in the models. First, it is becoming clear that the learned vectors of each model are different. It is also easy to see that the *null* and *intermediate* models share some structural similarities that can be attributed to the random nature of the walks on the lattice graph. On the contrary, nodes in the *real* model are demonstrating a more clustered nature, effectively revealing that people do not move in the city randomly, but rather following specific patterns of semantic similarity. Apparently, the various clusters of nodes in the visualization indicate that the areas represented by these nodes share some latent semantic similarity with each other.

While the t-SNE visualization is adequate for providing a summary of the embedded vectors, it doesn't provide information about pair-wise similarity of nodes. To address that the exploratory analysis relies on heat map visualization. Fig. 5.3a, 5.3b and 5.3c present examples of heat maps that showcase the pair-wise similarities of a specific node $u = 23$ to all other nodes in the analysis, for the three models, respectively. The example is chosen so as to demonstrate the differences of the models. It is clear from these heat maps that while node $u = 23$ reveals large semantic similarities with other nodes in the *real* model (depicted by darker colors), in the *null* model these similarities are much less stressed (represented by lighter colors). It is also interesting to see that our *intermediate* model is representing a middle situation between the *real* and *null* model, as expected. This is because in the *intermediate* model the origin node of every walk is the same as in the original walk in the real data.

5.1.4 Quantitative Analysis of Models

We are now ready to provide a more deep quantitative analysis of the models.

We start by the analysis of the pair-wise similarities of every pair of vectors that is learned for all three models (note that number of pairs of nodes are in the order of $\mathcal{O}(n^2)$, where n is the number of nodes). For each model, we rank the pairs based on their similarity score, in a descending order. Fig. 5.4 shows the results for all three models. It can be observed that (i) the *real* model is different than the *null* model and the *intermediate* model, respectively; (ii) the *real* model depicts a consistently lower similarity at the same level of rank compared to the *null* and the *intermediate* models, indicating that the similarity scores in these models are more well-distributed due to randomness. By focusing on the x-axis (rank of a pair of nodes) one can identify the pairs of nodes that depict the largest cosine similarity (leftmost), which represent geographic areas that are semantically similar.

Further, as a way to discover the most interesting pairs of nodes, we calculate the cosine similarity differences between the pairs of models – (real model vs intermediate model) and (real model vs null model). Fig. 5.5 shows the results. It can be seen that (i) for both cases there are a few only pairs of nodes that depict very high difference, indicating semantic relationships of high "interestingness"; (ii) the behavior for both comparisons is same, as indicated by the same trend; the slight variation can be explained by the way null models have been created. Note that for this experiment, we had to eliminate the pair of nodes whose similarity is less than zero, as we are considering cosine similarity in the positive space $[0, 1]$. Also, when we are reporting pairs of nodes, we only report on the pairs that can be defined in all three models.

In order to get a better understanding of how the different models compare to each other, for each model we construct a histogram that represents the distribution of the pair-wise similarity values in it. Fig. 5.6 shows the results for each model. We observe that (i) the *real* is different than the *null* model ($\chi^2 = 4.0854e + 05 \gg 0$) and the *intermediate* model ($\chi^2 = 3.0426e + 05 \gg 0$); (ii) the *real* has high concentration of smaller cosine similarity values (shifted on the left), while the *real* and *intermediate* models are well-distributed. Lastly, null model has a major chunk of pairs of nodes between 0.2 and 0.4 which can be attributed to the random nature of walks in this model.

5.2 Case Study II: City of Porto

5.2.1 Data

This dataset is based on $1,710,671$ trajectories of 442 taxis operating in the city of Porto, Portugal in a period from 01/07/2013 to 30/06/2014. For this study we rely on a random sample S that includes $10,000$ trajectories. Table 5.2 describes the features of dataset.

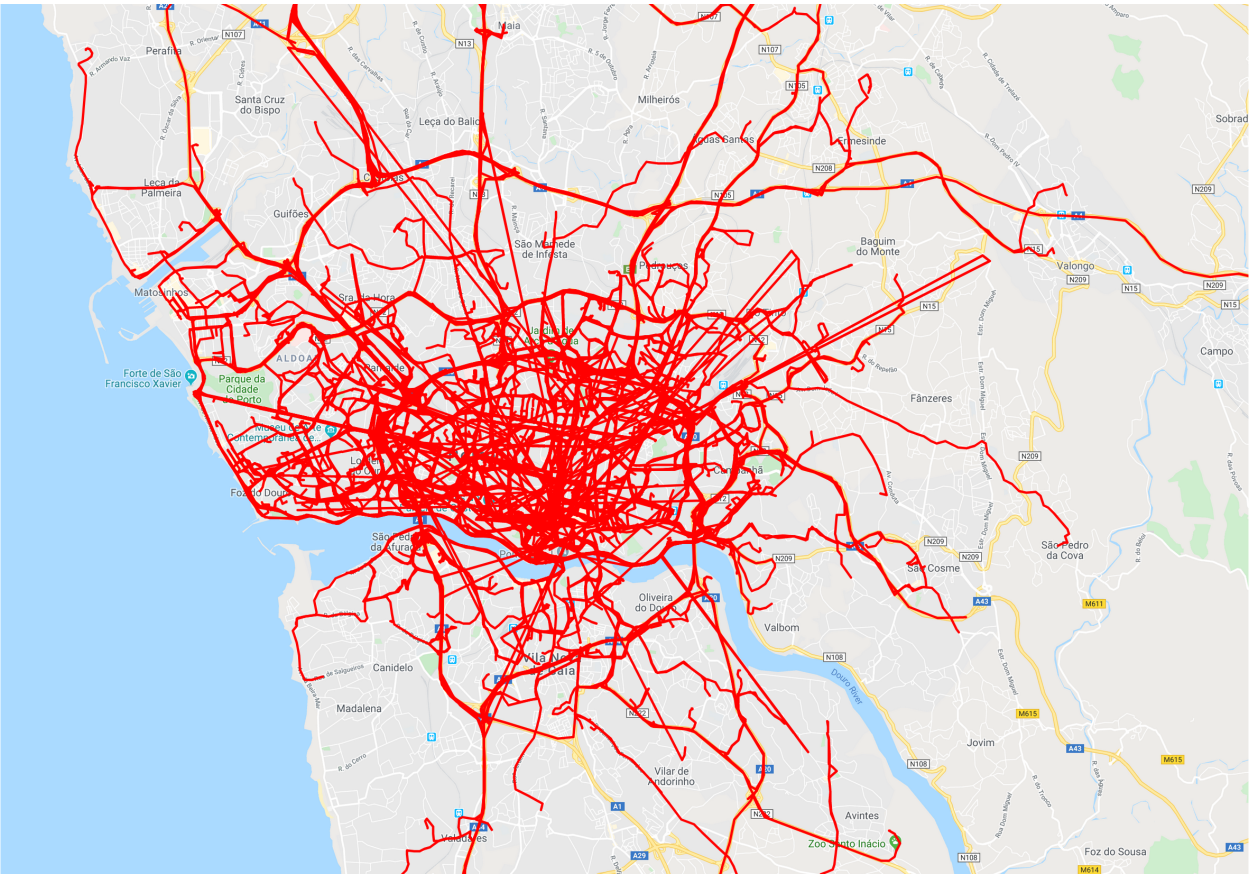


Figure 5.7: Sample trajectories of City of Porto taxi dataset.

Table 5.2: Dataset Features City of Porto

Columns	**Description**
trip_id	unique identifier for each trip
call_type	identifies the way used to demand this service
origin_call	unique identifier for each phone number
origin_stand	unique identifier for taxi stand
taxi_id	unique identifier for taxi driver
timestamp	identifies trips start
missing_data	it is FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing
polyline	it contains a list of GPS coordinates mapped as a string

5.2.2 Models

Given an observation area A defined by trajectories in the city of Porto, a set of trajectories S and a uniform grid $grid_{rc}$ with $r = 35$ rows and $c = 35$ columns, we use the methods described in Chapter 3 and 4 to obtain three models: the *real model*, the *null model* and the *intermediate model*.

5.2.3 Exploratory Analysis of Models

We begin the analysis of the models with an exploratory analysis. Similar to the previous case study, we first employ the t-SNE method to visually summarize the learned representations of the three models. Fig. 5.8a, 5.8b and 5.8c show the results. Again, it is becoming clear that the learned vectors of each model are different. It is also easy to see that the *null* and *intermediate* models share some structural similarities as in the NYC study. On the contrary, nodes in the *real* model are demonstrating a more clustered nature, indicating that the areas represented by these nodes share some latent semantic similarity with each other. In addition, Fig. 5.9a, 5.9b and 5.9c present examples of heat maps that aim to showcase the pair-wise similarities of a specific node $u = 189$ to all other nodes in the analysis, in the three models, respectively. The example is chosen so as to demonstrate the differences in the models. It is evident from these heat maps that while node $u = 189$ reveals large semantic similarities with other nodes in the *real* model (depicted by darker colors), in the *null* model these similarities are much less stressed (represented by lighter colors). Contrary to the case of NYC, it is interesting to see that the *intermediate* model is now representing a similar case to the *real* model.

5.2.4 Quantitative Analysis of Models

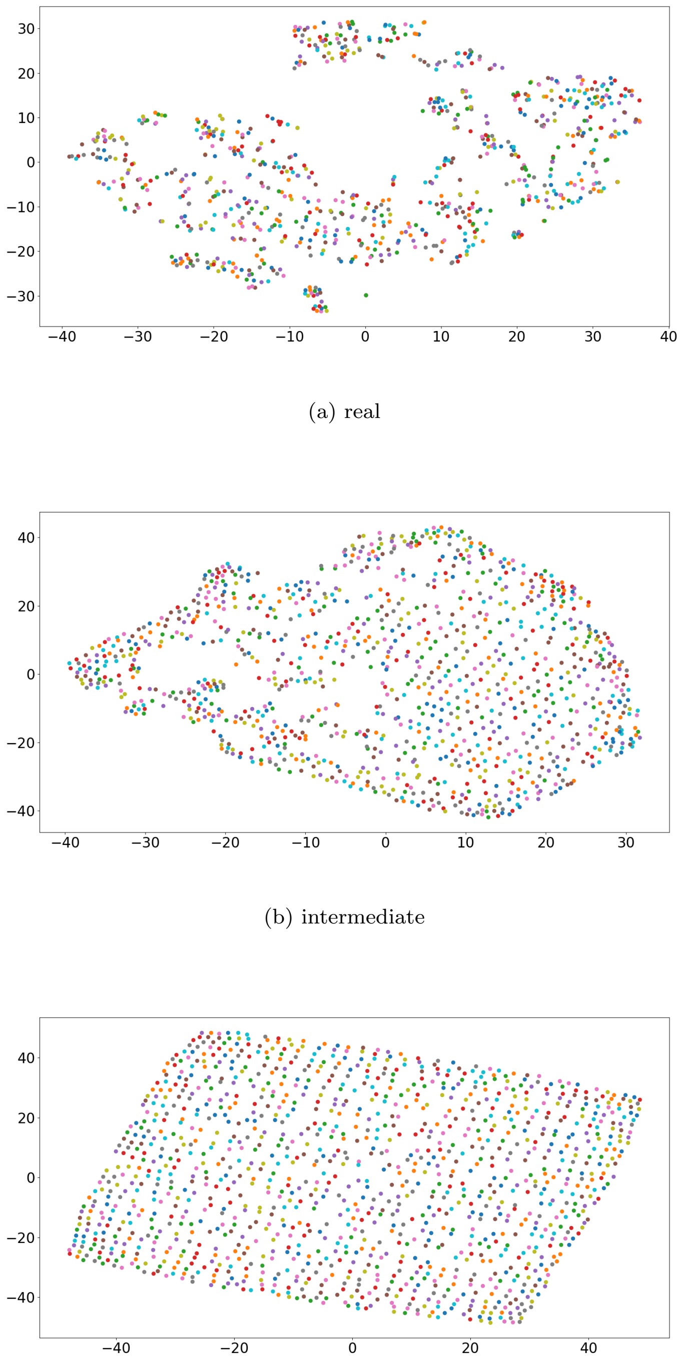
We are now ready to provide a more deep quantitative analysis of the models.

We begin by analysis of the pair-wise similarities. For each model, we rank the pairs based on their similarity score, in a descending order. Fig. 5.10 shows the results for all three models. Similarly to the NYC study, we observe that the *real* is different than the *null* and *intermediate* models, respectively.

We also calculate the cosine similarity differences between the pairs of models – (*real* vs *intermediate*) and (*real* vs *null*). Fig. 5.11 shows the results. Similarly to the NYC study, it can be seen that (i) for both cases there are a few only pairs of nodes that depict very high difference, indicating semantic relationships of high "interestingness"; (ii) the behavior for both comparisons is same, as indicated by the same line trends.

In order to get a better understanding of how the different models compare to each other, for each model we construct a histogram that represents the distribution of the pair-wise similarity values in it. Fig. 5.12 shows the results for each model. We observe that (i) the *real* model is different than the *null* model ($\chi^2 = 6.1697e + 05 \gg 0$) and the *intermediate* model ($\chi^2 = 7.8492e + 05 \gg 0$); (ii) the *real* has high concentration of smaller cosine similarity scores (shifted on the left), while the *real* and *intermediate* models are more well-distributed.

Our statistical method and analysis has concluded that both in the case of NYC and the City of Porto the *real* model is significantly different than both the null and *intermediate* model, which means that the null hypothesis (that people in the City of New York and Porto move randomly) can be rejected.

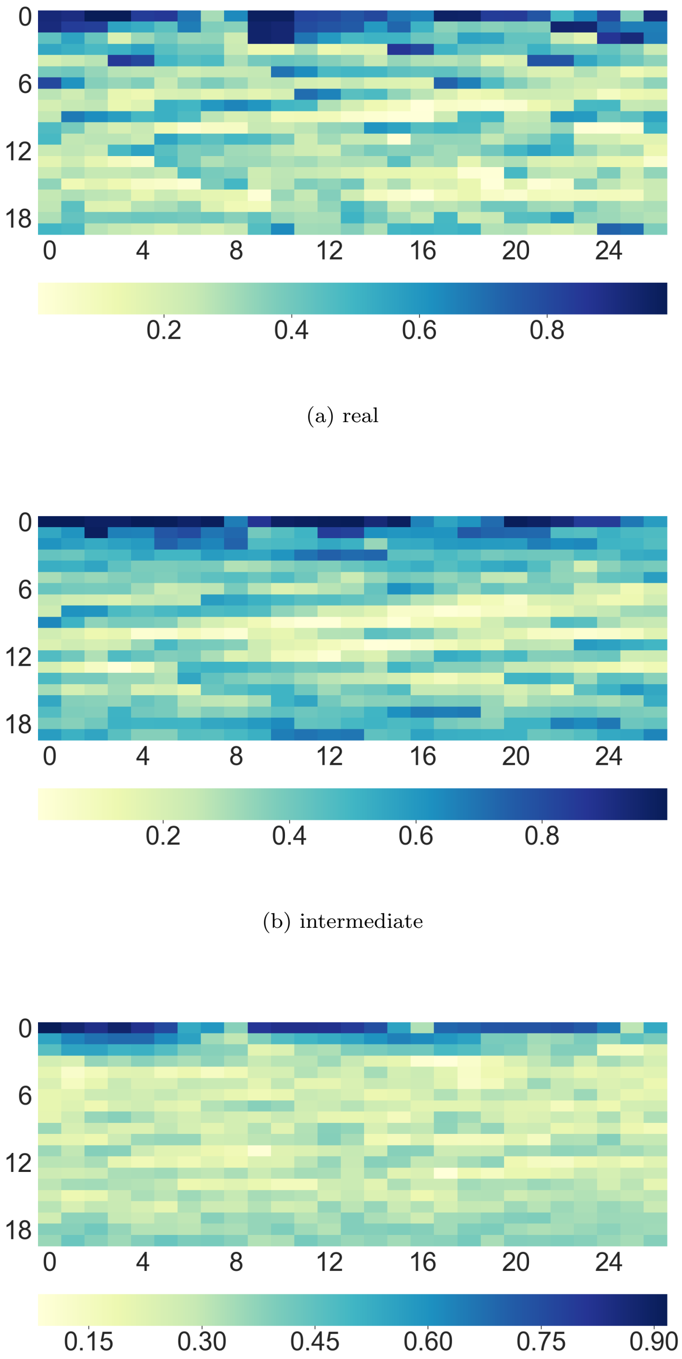


(a) real

(b) intermediate

(c) null

Figure 5.2: Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models).



(a) real

(b) intermediate

(c) null

Figure 5.3: Exploratory Analysis of Models. (a), (b) and (c) provide the pair-wise similarities of a specific node $u = 23$ to all other nodes in the three models using heat maps (it allows for a one-to-many comparisons of models).

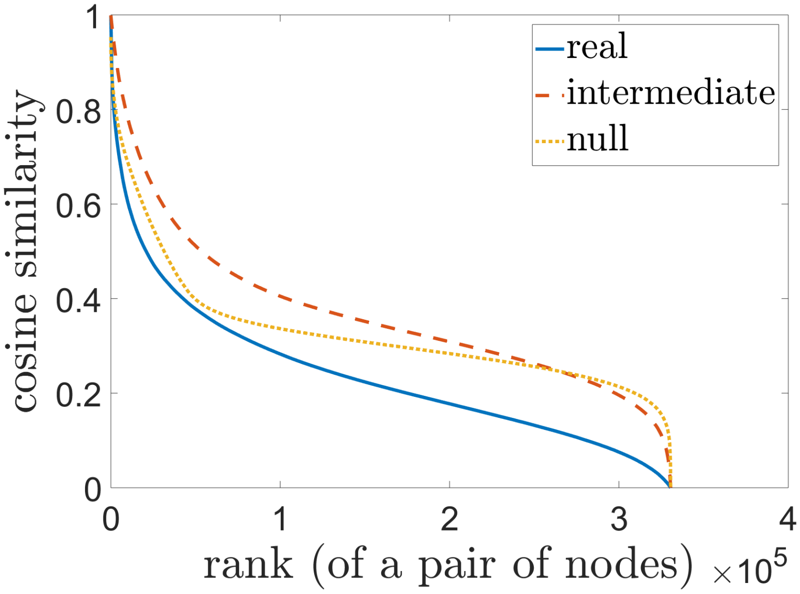


Figure 5.4: Cosine similarity between pairs of nodes.

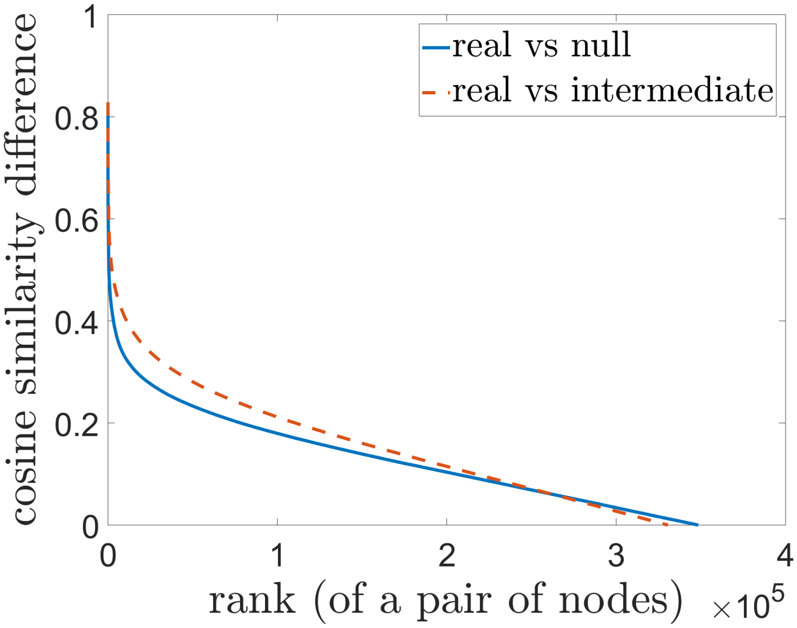
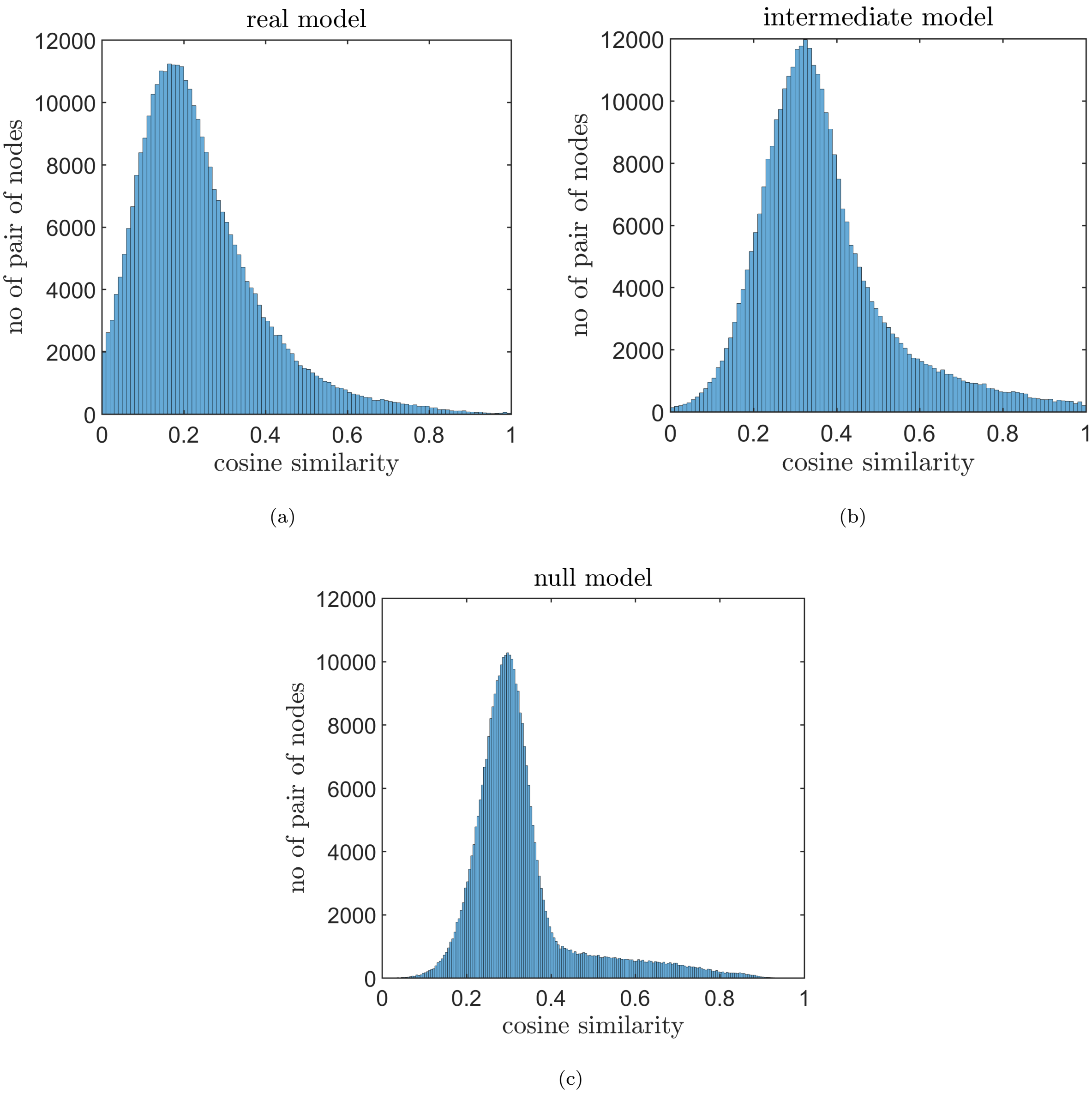


Figure 5.5: Cosine similarity differences between pair of nodes.

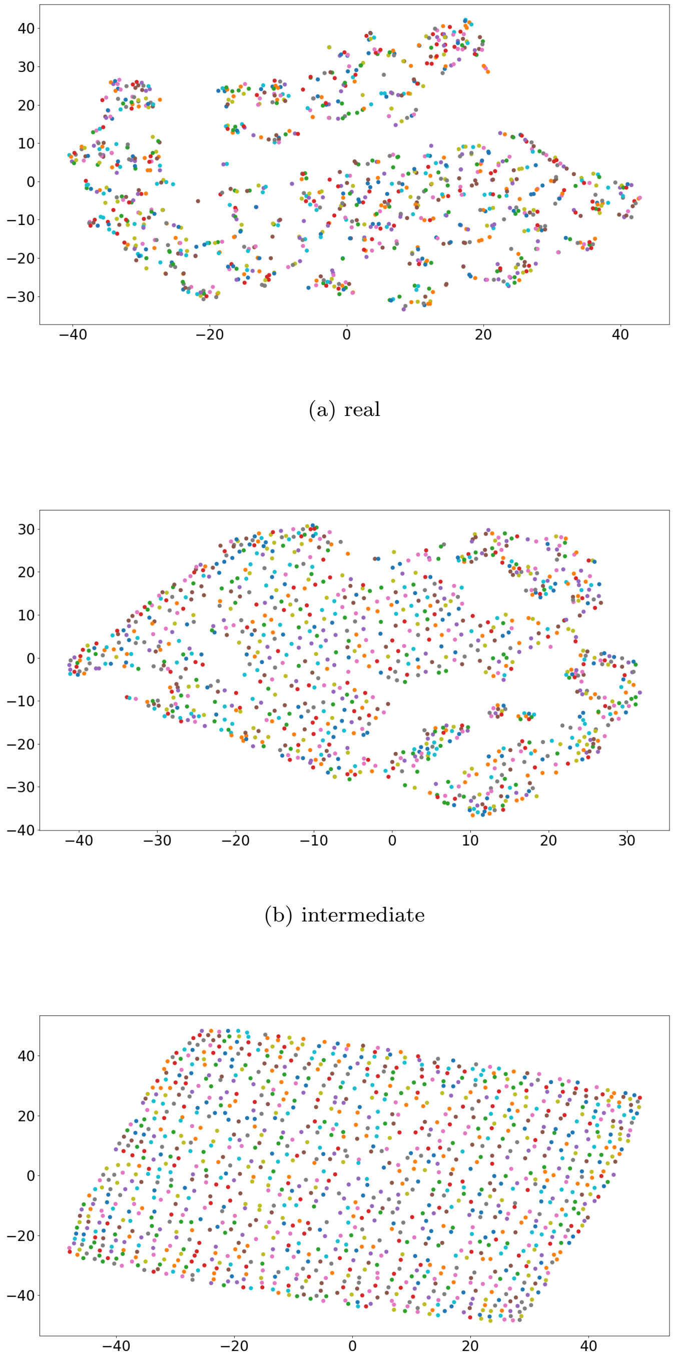


(a)

(b)

(c)

Figure 5.6: Distribution of cosine similarity values between pair of nodes.

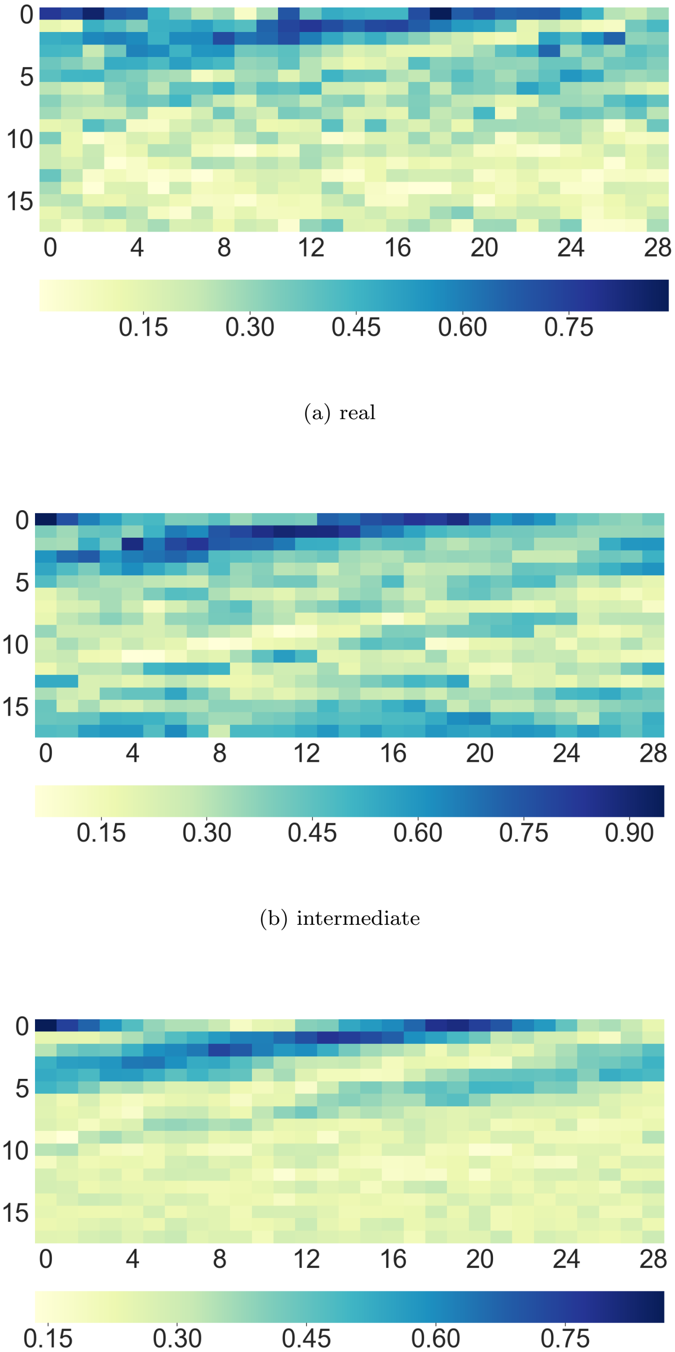


(a) real

(b) intermediate

(c) null

Figure 5.8: Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models).



(a) real

(b) intermediate

(c) null

Figure 5.9: Exploratory Analysis of Models. (a), (b) and (c) provide the pair-wise similarities of a specific node $u = 189$ to all other nodes in the three models using heat maps (it allows for a one-to-many comparisons of models).

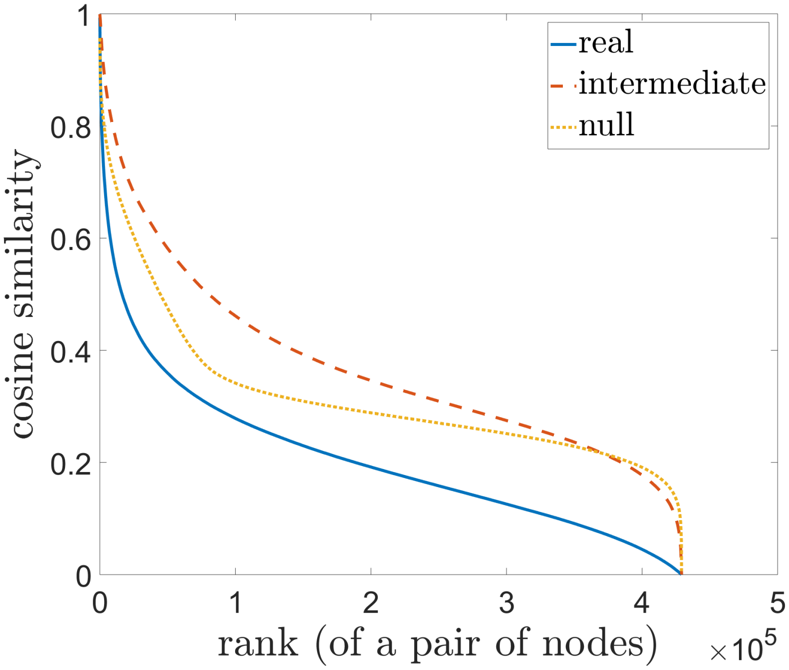


Figure 5.10: Cosine similarity between pairs of nodes.

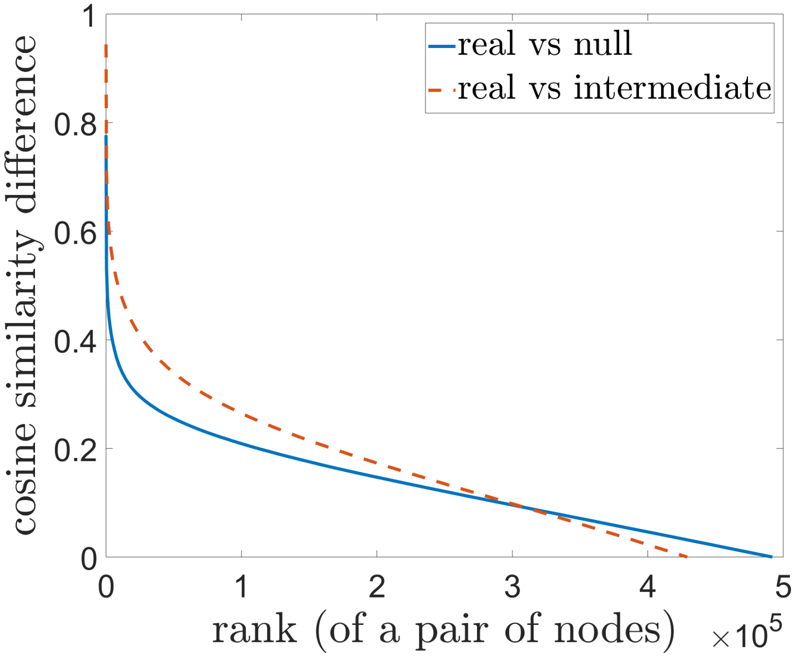
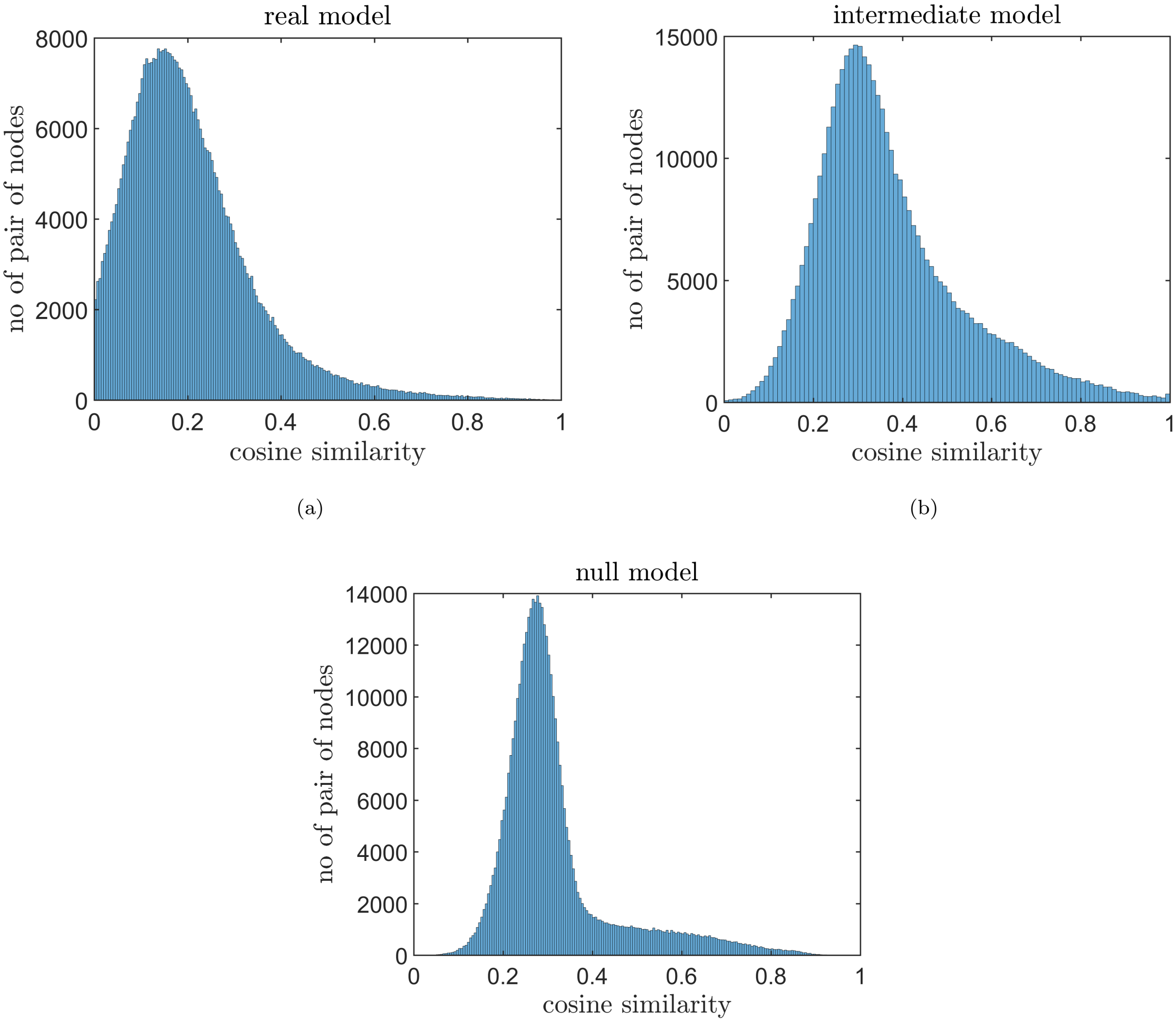


Figure 5.11: Cosine similarity differences between pair of nodes.



(a)

(b)

(c)

Figure 5.12: Distribution of cosine similarity values between pair of nodes.

6 System Implementation

This chapter briefly mention some of the existing tools we use towards utilizing trajectories and points-of-interest datasets to be able to come with the analysis presented in the previous chapter.

6.1 Spatial Database

PostgreSQL[2] is an open source object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. It extends the SQL language combined with many features that safely store and scale the most complicated data workloads. PostGIS [3] is a spatial database extender for PostgreSQL. It adds spatial functions such as distance, area, union, intersection, and geometry data types to PostgreSQL.

Among the reasons to use PostgreSQL and PostGIS were primarily focused to run spatial queries on the acquired real world datasets coming from NYC and the City of Porto. From trajectories T to random walks W_{v_i} and translating them further into POIs P creates an overhead in terms of computation. Frameworks such as PostgreSQL and PostGIS are designed to work efficiently with spatial data sets.

6.1.1 Modeling of Datasets in Database

To store data in the database we had to change it to Well-known text (WKT) format which is a text markup language for representing vector geometry objects on a map. A binary equivalent, known as well-known

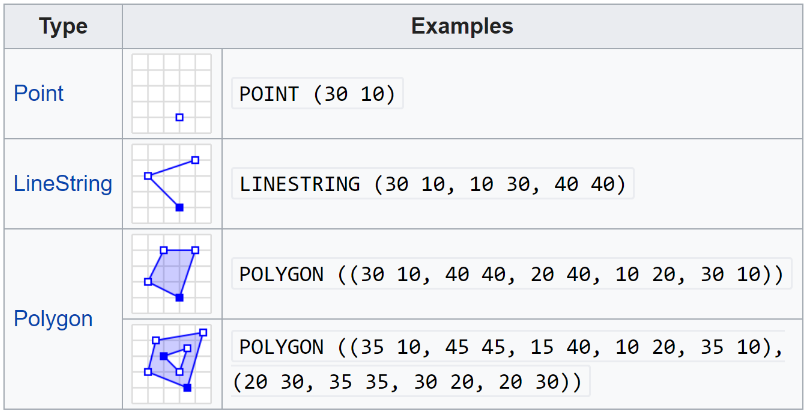
[2] https://www.postgresql.org/about/

[3] https://postgis.net/

binary (WKB), is used to transfer and store the same information on databases. WKT can represent the following distinct geometric objects:

- Point, MultiPoint
- LineString, MultiLineString
- Polygon, MultiPolygon, Triangle
- PolyhedralSurface
- TIN (Triangulated irregular network)
- GeometryCollection

We are using Point, LineString and Polygon objects to store our datasets i.e., Point is being used to store POI datasets, LineString is used to store trajectory datasets and similarly Polygon is used to store grid cells coordinates. An example of these geometric objects is shown in the (Fig 6.1)



Type	Examples	
Point		`POINT (30 10)`
LineString		`LINESTRING (30 10, 10 30, 40 40)`
Polygon		`POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))`

		`POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10),`
		`(20 30, 35 35, 30 20, 20 30))`

Figure 6.1: Geometry primitives (2D) - PostGIS

Point A spatial point represents a single location on the earth. It can be represented by a single coordinate such as (x, y, z) where x, y, z are latitude, longitude and altitude. Points are used to represent objects when the exact details, such as shape or size are not important. For example. cities on a map of the world can be described as points, while a map of a single state might represent cities as polygons.

LineString A linestring is a path from a *starting point* to an *ending point.* Generally roads and rivers are represented as linestrings.

Polygon A polygon represents an area. Generally polygons are used to represent objects whose size and shape are important. For example, city limits, parks, building footprints or bodies of water are all represented as polygons. We are using them to represent boundaries of grid cells i.e., geographical areas created by dividing observation area A.

6.1.2 Spatial Relationships

Spatial databases are powerful because they can store geometric objects and also have the ability to compare relationships between geometries. Questions such as "What is the closest subway station from my location?" or "Which sushi place is famous in downtown?" can only be answered by comparing geometries representing points-of-interests and object movement trajectories.

PostGIS provides set of methods to compare geometries. We are using following methods:

- ST_SRID
- ST_AsText
- ST_LineLocatePoint
- ST_Centroid
- ST_Dump
- ST_Intersection

ST_SRID A Spatial Reference System Identifier (SRID) is a unique value used to unambiguously identify projected, unprojected, and local spatial coordinate system definitions. These coordinate systems form the heart of all GIS applications. We are using it to assign 4326 EPSG references[4] to Points, LineStrings and Polygons. For Example, following statement assigns SRID to points geometry.

```
update poi

set geom_point = st_setsrid(geom_point, 4326)
```

where *poi* is the table that contains POIs and *geom_point* is a geometry type column containing coordinates for each POINT geometric object. We cannot perform spatial queries between geometric objects without assigning SRID to their coordinates.

ST_AsText

```
ST_AsText(geometry g1);
```

This method returns the Well-Known Text (WKT) representation of the geometry/geography without SRID metadata. We are using it to understand results after performing geometric operations.

ST_LineLocatePoint

```
ST_LineLocatePoint(geometry a_linestring, geometry a_point);
```

Finds the point on a given linestring that is closest to a given point.

ST_Centroid

```
geometry ST_Centroid(geometry g1);
```

Computes the geometric centre of a geometry, or equivalently, the centre of mass of the geometry as a Point.

ST_Dump

```
geometry_dump[] ST_Dump(geometry g1);
```

Returns a set of geometry_dump rows, formed by a geometry (geom) and an array of integers (path).

[4] http://spatialreference.org/

ST_Intersection

```
geometry ST_Intersection(geometry geomA , geometry geomB);
```

Returns the portion of geometry A and geometry B that is shared between the two geometries. The above methods can be understood by looking at the following query created to fetch cells c_{ij} through which each trajectory t_i is passing in the order of traversal i.e, c_{02}, c_{03}, ..., c_{67} etc.

```
WITH t1 AS (
    SELECT tr.traj_id, ce.cell_id,
    ST_LineLocatePoint(
      tr.traj_path,
      ST_CENTROID(
          (ST_DUMP(
              ST_Intersection(ce.coordinates, tr.traj_path)
          )).geom
      )
  ) AS distance
  FROM cells ce, traj tr
),
t2 AS (
  SELECT t1.traj_id, t1.cell_id,
  COALESCE(LEAD(t1.cell_id) OVER(ORDER BY t1.traj_id, t1.distance), -1) AS next_cell_id
  FROM t1
)
SELECT t2.traj_id, t2.cell_id into table traj_as_cells_porto
FROM t2
WHERE t2.cell_id <> t2.next_cell_id;
```

Where **Coalesce** return the first non-null value, **Lead** provides access to a row at a specified physical offset which follows the current row and **Over** defines user-specified set of rows within a query result.

6.2 Environment

For the rest of analytical tasks i.e., trajectory visualization, grid construction, lattice graph construction, vector embeddings we used *Python 3.6.4* version with the following libraries *networkx, pandas, numpy, seaborn, matplotlib* etc. Plotting comparisons between vector cosine similarity and differences and histograms were implemented using *Matlab.* Case study experiments were performed on a PC with 64 GB of memory with Intel(R) Core(TM) i7-7700 CPU @ 2x3.60GHz & 4 cores.

7 Conclusion

The main objective of this research was to leverage the abundance of trajectory data available to accurately learn latent relationships between different geographical areas (e.g., semantically correlated neighborhoods of a city), in an unsupervised fashion. To address this problem we first employed state-of-the-art deep learning methods, including methods of network representation learning. These methods allow to learn low-dimensional representations of geographical areas by treating trajectories as random walks on a grid network. As a result, we were able to design a method for learning low-dimensional representations of the nodes of a lattice graph, each of which represent a geographical area of the observation space. These representations can then be used to efficiently mine relationships between the geographical areas. This is important, as it allows to inform applications and services in various domains, ranging from location-based services such as points-of-interest recommendations, to finding relationships between different parts of a city as revealed by patterns in trajectory data. In addition, we designed and evaluated a statistical method that allows to compare the learned representations to a theoretical null model. More importantly, we demonstrated that since the method is based on learning embeddings of the geographical areas in the same low-dimensional space, it allows to inform a comparative analysis between different observation areas (e.g., different cities). To our knowledge, this is the first attempt to employ a well-defined statistical method to distinguish geographical proximity to semantic proximity by operating only on input dataset of raw trajectories. We demonstrated the effectiveness and usefulness of the proposed embedding and statistical method in two case studies utilizing real-world data coming from the New York City and the city of Porto, Portugal, respectively. Overall, this analysis can improve our understanding of how space is perceived by individuals and inform better decisions of urban planning. The methods we described are generic and can probably be easily adopted in similar studies.

7.1 Future Work

Understanding geographical space including object interaction with points-of-interest leads to an interesting discovery and knowledge that could be implemented as a future continuation of this work. The knowledge we gain can help us in developing expert profiles. These profiles can be built automatically by identifying and linking trajectories to specific objects and further assigning them expertise based on their interaction with points-of-interest and geographical space they are traversing. Simply put, these profiles can be used for various question answering tasks and enhance user experience of visiting a new geographical space.

Bibliography

[1] Anagnostopoulos, A., Kumar, R., and Mahdian, M. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), pp. 7–15.

[2] Andersen, R., Chung, F., and Lang, K. Local graph partitioning using pagerank vectors. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* (2006), IEEE, pp. 475–486.

[3] Bao, J., He, T., Ruan, S., Li, Y., and Zheng, Y. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the 23rd ACM SIGKDD* (2017), ACM, pp. 1377–1386.

[4] Cho, E., Myers, S. A., and Leskovec, J. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD* (2011), ACM, pp. 1082–1090.

[5] Christoforidis, G., Kefalas, P., Papadopoulos, A., and Manolopoulos, Y. Recommendation of points-of-interest using graph embeddings. In *2018 IEEE 5th Int. Conf. on DSAA* (2018), IEEE, pp. 31–40.

[6] Dodge, S., Weibel, R., and Forootan, E. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems 33*, 6 (2009), 419–434.

[7] Fouss, F., Pirotte, A., Renders, J.-M., and Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering 19*, 3 (2007), 355–369.

[8] Furletti, B., Cintia, P., Renso, C., and Spinsanti, L. Inferring human activities from gps tracks. In *KDD* (2013), ACM, p. 5.

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

[10] Guo, D., Liu, S., and Jin, H. A graph-based approach to vehicle trajectory analysis. *Journal of Location Based Services 4*, 3-4 (2010), 183–199.

[11] Hang, M., Pytlarz, I., and Neville, J. Exploring student check-in behavior for improved point-of-interest prediction. In *Proceedings of the 24th ACM SIGKDD* (2018), ACM, pp. 321–330.

[12] Heidari, F., and Papagelis, M. Evonrl: Evolving network representation learning based on random walks. In *International Conference on Complex Networks and their Applications* (2018), Springer, pp. 457–469.

[13] Heidari, F., and Papagelis, M. Evolving network representation learning based on random walks. *Applied Network Science 5*, 1 (2020), 1–38.

[14] Hoff, P. D., Raftery, A. E., and Handcock, M. S. Latent space approaches to social network analysis. *Journal of the american Statistical association 97*, 460 (2002), 1090–1098.

[15] Kucuk, A., Hamdi, S. M., Aydin, B., Schuh, M. A., and Angryk, R. A. Pg-trajectory: A postgresql/postgis based data model for spatiotemporal trajectories. In *BDCloud-SocialCom-SustainCom* (2016), pp. 81–88.

[16] Kumar, S., Zhang, X., and Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD* (2019), pp. 1269–1278.

[17] Lappas, T., Liu, K., and Terzi, E. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD* (2009), ACM, pp. 467–476.

[18] Lee, J.-G., Han, J., and Whang, K.-Y. Trajectory clustering: a partition-and-group framework. In *SIGMOD* (2007), pp. 593–604.

[19] Li, H., Ge, Y., Hong, R., and Zhu, H. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD* (2016), ACM, pp. 975–984.

[20] Lin, S., Hong, W., Wang, D., and Li, T. A survey on expert finding techniques. *Journal of Intelligent Information Systems 49*, 2 (2017), 255–279.

[21] Liu, B., Fu, Y., Yao, Z., and Xiong, H. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD* (2013), ACM, pp. 1043–1051.

[22] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605.

[23] Mehmood, S., and Papagelis, M. Learning semantic relationships of geographical areas based on trajectories. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)* (2020), IEEE, p. In Press.

[24] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *NIPS* (2013), pp. 3111–3119.

[25] Papagelis, M., Murdock, V., and van Zwol, R. Individual behavior and social influence in online social systems. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia* (2011), pp. 241–250.

[26] Pechlivanoglou, T., and Papagelis, M. Fast and accurate mining of node importance in trajectory networks. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 781–790.

[27] Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *KDD* (2014), ACM, pp. 701–710.

[28] Sawas, A., Abuolaim, A., Afifi, M., and Papagelis, M. Tensor methods for group pattern discovery of pedestrian trajectories. In *2018 19Th IEEE International Conference on Mobile Data Management (MDM)* (2018), IEEE, pp. 76–85.

[29] Sawas, A., Abuolaim, A., Afifi, M., and Papagelis, M. Trajectolizer: Interactive analysis and exploration of trajectory group dynamics. In *2018 19th IEEE International Conference on Mobile Data Management (MDM)* (2018), IEEE, pp. 286–287.

[30] Sawas, A., Abuolaim, A., Afifi, M., and Papagelis, M. A versatile computational framework for group pattern mining of pedestrian trajectories. *GeoInformatica 23*, 4 (2019), 501–531.

[31] Shang, S., Chen, L., Zheng, K., Jensen, C. S., Wei, Z., and Kalnis, P. Parallel trajectory-to-location join. *IEEE TKDE* (2018).

[32] Siła-Nowicka, K., Vandrol, J., Oshan, T., Long, J. A., Demšar, U., and Fotheringham, A. S. Analysis of human mobility patterns from gps trajectories and contextual information. *International Journal of Geographical Information Science 30*, 5 (2016), 881–906.

[33] Tobler, W. R. A computer movie simulating urban growth in the detroit region. *Economic geography 46*, sup1 (1970), 234–240.

[34] Toohey, K., and Duckham, M. Trajectory similarity measures. *Sigspatial Special 7*, 1 (2015), 43–50.

[35] van Kreveld, M., and Luo, J. The definition and computation of trajectory and subtrajectory similarity. In *Proc. of the 15th annual ACM symposium on Advances in GIS* (2007), pp. 1–4.

[36] Wang, H., Fu, Y., Wang, Q., Yin, H., Du, C., and Xiong, H. A location-sentiment-aware recommender system for both home-town and out-of-town users. In *Proceedings of the 23rd ACM SIGKDD* (2017), ACM, pp. 1135–1143.

[37] Wu, H.-R., Yeh, M.-Y., and Chen, M.-S. Profiling moving objects by dividing and clustering trajectories spatiotemporally. *IEEE Trans. on Knowledge and Data Engineering 25*, 11 (2012), 2615–2628.

[38] Wu, R., Luo, G., Shao, J., Tian, L., and Peng, C. Location prediction on trajectory data: A review. *BDMA 1*, 2 (2018), 108–127.

[39] Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., and Wang, S. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016), ACM, pp. 15–24.

[40] Xitong Zhang, Liyang Xie, Z. W., and Zhou, J. Boosted trajectory calibration for traffic state estimation. *IEEE Conference on Data Mining* (2019).

[41] Xu, Y., Zhou, D., and Lawless, S. Inferring your expertise from twitter: Integrating sentiment and topic relatedness. In *2016 IEEE/WIC/ACM International Conference on (WI)* (2016), IEEE, pp. 121–128.

[42] Yuan, G., Sun, P., Zhao, J., Li, D., and Wang, C. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review 47*, 1 (2017), 123–144.

[43] Zanlungo, F., Ikeda, T., and Kanda, T. Potential for the dynamics of pedestrians in a socially interacting group. *Physical Review E 89*, 1 (2014), 012811.

[44] Zheng, Y. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST) 6*, 3 (2015), 29.

[45] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing* (2008), ACM, pp. 312–321.

[46] Zheng, Y., Xie, X., Ma, W.-Y., et al. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull. 33*, 2 (2010), 32–39.

[47] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web* (2009), ACM, pp. 791–800.

[48] Zhou, F., Yue, X., Trajcevski, G., Zhong, T., and Zhang, K. Context-aware variational trajectory encoding and human mobility inference. In *The World Wide Web Conference* (2019), pp. 3469–3475.