

Learning Semantic Relationships of Geographical Areas based on Trajectories

Saim Mehmood and Manos Papangelis

Lassonde School of Engineering, York University, Toronto, Canada

{saim, papaggel}@eecs.yorku.ca

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 paper, 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 our 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.

Index Terms—trajectory data mining, network representation learning, spatial databases, machine learning, statistical inference

I. 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 bird’s 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 [1], [2], trajectory clustering and outlier detection [3], [4], or crowd behavioral analysis [5]–[9]. A comprehensive survey of classical trajectory data mining can be found in [10]. Recent advances on trajectory data mining look on *network dynamics of trajectories*, such as mining group patterns of trajectories [11] and mining the importance of a moving object in trajectory networks [12].

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 [13]–[15]. 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. 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 of their city? To what extent 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 extent 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.

The remainder of this paper is organized as follows: Section II introduces notation and presents our method for learning semantic relations of geographical areas. Section III formally presents the statistical model. In Section IV we present two real-world case studies to demonstrate how the model can be applied in practice. After reviewing the related work in Section V, we conclude in Section VI.

II. LEARNING SEMANTIC RELATIONS OF GEOGRAPHICAL AREAS

In this section we describe a method that given a set of trajectories $T = \{t_0, t_1, t_2, \dots, t_i, \dots, t_{n-1}\}$, 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 1 shows the steps of the proposed method and Table I summarizes important notation.

A. Construction of a Uniform Grid

Let a trajectory $t = \{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$ 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:

$$t = \{p_0, p_1, p_2, \dots, p_n\} \quad (1)$$

where p_{i-1} is the i th point of the trajectory t . While individual trajectories of moving objects are defined at a lower level of

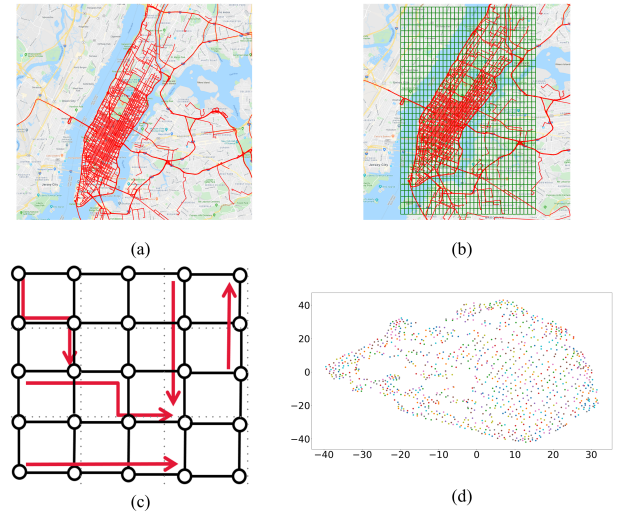


Fig. 1: Illustration of the proposed method for learning semantic relationships of geographical areas: (a) sample trajectories of the New York City taxi dataset, (b) trajectories traversing grid cells of a uniform grid, (c) trajectories as random walks on a lattice graph, (d) node embeddings of the lattice graph.

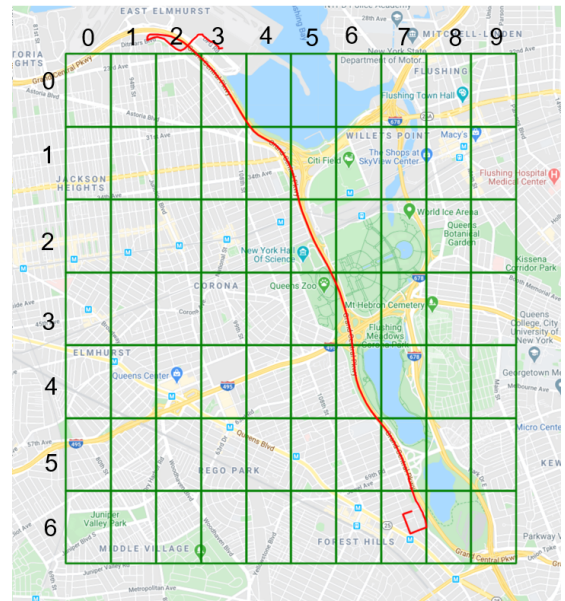


Fig. 2: Sample grid and sample trajectory over grid cells.

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:

$$grid_{rc} = \{c_{00}, c_{01}, c_{10}, \dots, c_{r-1c-1}\} \quad (2)$$

TABLE I: Summary of Notations

| Symbol | Description |
|----------------|--|
| T | set of trajectories $T = \{t_0, t_1, t_2, \dots, t_i, \dots, t_n\}$ |
| A | observation space |
| $grid_{rc}$ | a uniform grid of r rows and c columns $grid_{rc} = \{c_{00}, c_{01}, c_{10}, \dots, c_{ij}, \dots, c_{rc}\}$ |
| $n_{c_{ij}}$ | representing all grid cells c_{ij} |
| c_{ij} | the grid cell at the i th row and j th 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 |

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 a sequence of grid cells on $grid_{rc}$. For example, Fig. 2 shows a 7×10 grid and a sample trajectory that traverses 12 grid cells, starting at c_{02} moving to $c_{03}, c_{04}, c_{14}, c_{15}, \dots$, 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.

B. 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 .

C. 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 of that is that random walks on a graph have been successfully used as a way to obtain semantic relationships between nodes of a graph [16], [17]. Therefore, we can 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 $W_{v_i}^1, W_{v_i}^2, \dots, W_{v_i}^k$ 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 [18] [19], in different kinds of networks. Nodes in a network can be classified on the basis of *homophily* and *structural equivalence* [20] 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. 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 2, the grid cells $c_{00}, c_{10}, \dots, c_{61}$ that a trajectory traverses have the same structural role, as they are all part of the same trajectory.

D. 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 [16]. Using that approximation objective and the skip-gram model of *node2vec* [17], we obtain embeddings that are optimized by stochastic gradient decent so that:

$$Pr(v_j | \mathbf{v}_i) \propto \exp(\mathbf{v}_j^T \mathbf{v}_i) \quad (3)$$

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 model¹. We set our window size to $w = 10$ and the number of dimensions to $d = 128$. A similar approach has been employed in [21] 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 [22] 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 far they are from each other. To achieve that, we rely on generating m random permutations of a single trajectory

¹<https://github.com/RaRe-Technologies/gensim>

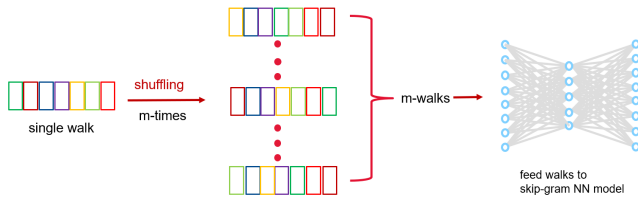


Fig. 3: Generating m walks from a single walk based on trajectory permutations and feeding them to skip-gram.

and providing these m trajectories to the skip-gram model (as shown in Fig 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 $W_{v_i} = \{v_i, \dots, v_k\}$ 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.

III. 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?). Our 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 [23], [24].

A. 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 $G(V, E)$.

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.

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.

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.

B. 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 allow to test 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.

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.

a) *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{v}_i and \vec{v}_j of nodes i and j , their cosine similarity is given by:

$$\cos\theta(\vec{v}_i, \vec{v}_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{\|\vec{v}_i\| \cdot \|\vec{v}_j\|} = \frac{\sum_1^n \mathbf{v}_i \mathbf{v}_j}{\sqrt{\sum_1^n \mathbf{v}_i^2} \sqrt{\sum_1^n \mathbf{v}_j^2}} \quad (4)$$

where $\vec{v}_i \cdot \vec{v}_j = \sum_1^n \mathbf{v}_i \mathbf{v}_j = \mathbf{v}_{i1} \mathbf{v}_{j1} + \mathbf{v}_{i2} \mathbf{v}_{j2} + \dots + \mathbf{v}_{in} \mathbf{v}_{jn}$ 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 are ignoring the pairs of nodes whose cosine similarity is negative. 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 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),

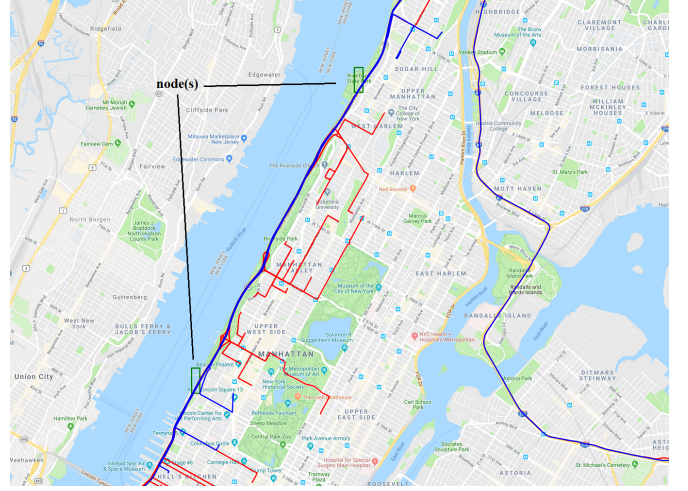


Fig. 4: Two grid cells (green rectangles) with several trajectories traversing both of them (shown as blue line), compared to trajectories that traverse either of them (shown as red line).

compared to trajectories that are traversing through either of the grid cells (shown as red lines).

b) *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:

$$d_{X,Y}(\vec{v}_i, \vec{v}_j) = |\cos\theta_X(\vec{v}_i, \vec{v}_j) - \cos\theta_Y(\vec{v}_i, \vec{v}_j)| \quad (5)$$

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., $d_{X,Y}(\vec{v}_i, \vec{v}_j) \geq \lambda_b$), 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).

c) *Distribution of Pair-wise Similarities*: We are interested to compare 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:

$$n = \sum_{i=1}^b m_i \quad (6)$$

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:

$$M_i = \sum_{j=1}^i m_j \quad (7)$$

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 Kolmogorov–Smirnov test statistic. For simplicity, we employ a chi-square distance:

$$\chi^2 = d(H^A, H^B) = \sum_{i=1}^b \frac{(H_i^A - H_i^B)^2}{H_i^A} \quad (8)$$

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

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.

a) *Model Embeddings (Many-to-many Visualisation)*: T-distributed Stochastic Neighbor Embedding (t-SNE) [25] 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.

IV. 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.

A. Case Study I: New York City (NYC)

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.

Fields containing pickup and dropoff points as pairs of (longitude, latitudes) coordinates. For the needs of our study we rely on a uniform random sample S that includes 10,000 trajectories. For each pair of pickup and dropoff locations, we utilize the Google Directions API to create actual trajectories in NYC.

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 Section II and III to obtain three models: the *real model*, the *null model* and the *intermediate model*.

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. 5a, 5b, and 5c 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.

4) *Quantitative Analysis of Models*: We are now ready to provide a more deep quantitative analysis of the models.

We start by 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. 6 shows the results for all three models. We observe 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.

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. 7 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” i.e., these points are above $\lambda_b = 0.5$ (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

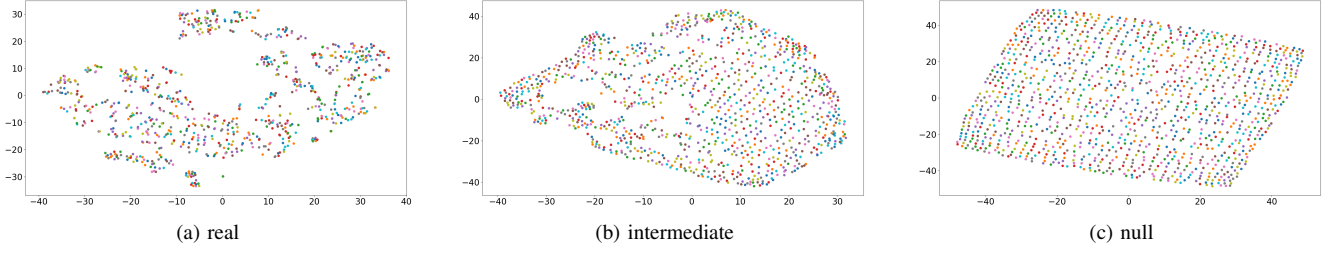


Fig. 5: Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models)

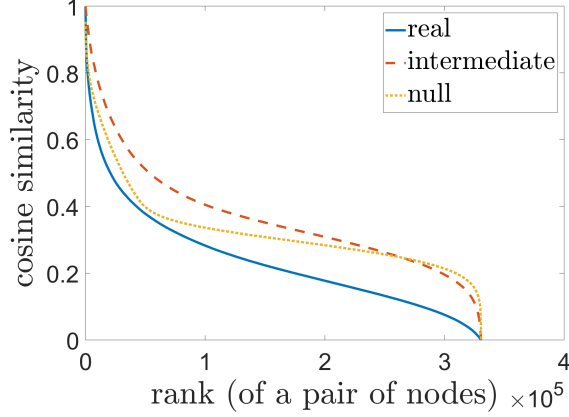


Fig. 6: Cosine similarity between pairs of nodes.

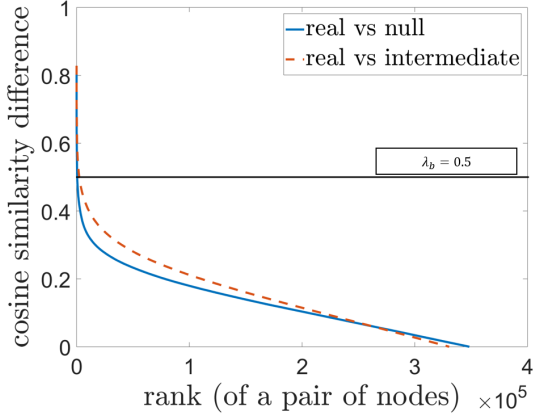


Fig. 7: Cosine similarity differences between pair of nodes.

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. 8 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), and both the *real* and *intermediate* models are well-distributed.

B. Case Study II: City of Porto

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 our study we rely on a random sample S that includes 10,000 trajectories.

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 Section II and III to obtain three models: the *real model*, the *null model* and the *intermediate model*.

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. 9a, 9b and 9c 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.

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. 10 shows the results for all three models. Similar 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. 11 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” i.e., these points are above $\lambda_b = 0.5$ (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. 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

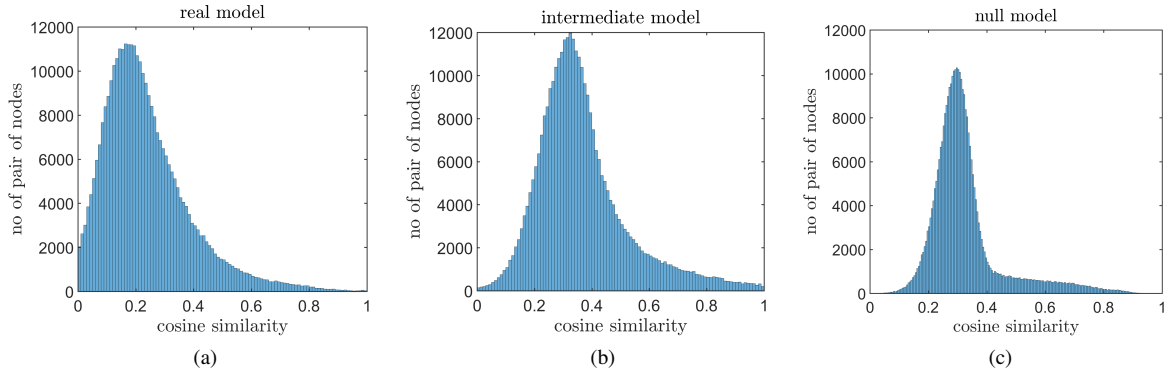


Fig. 8: Distribution of cosine similarity values between pair of nodes

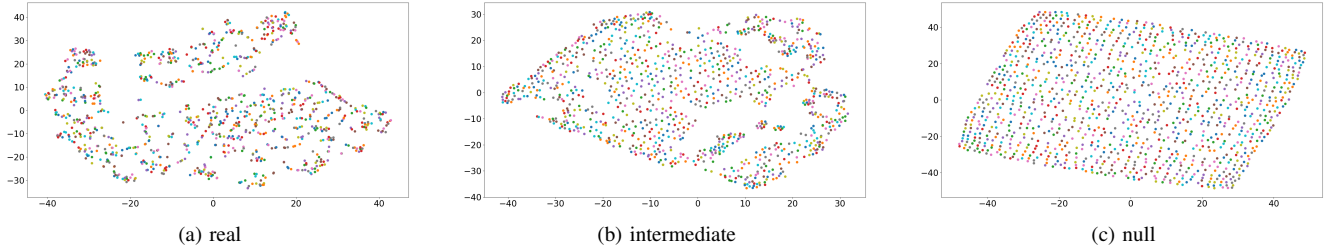


Fig. 9: Exploratory Analysis of Models. (a), (b) and (c) provide a summary visualization of learned vector embeddings using t-SNE (it allows for a many-to-many comparison of models).

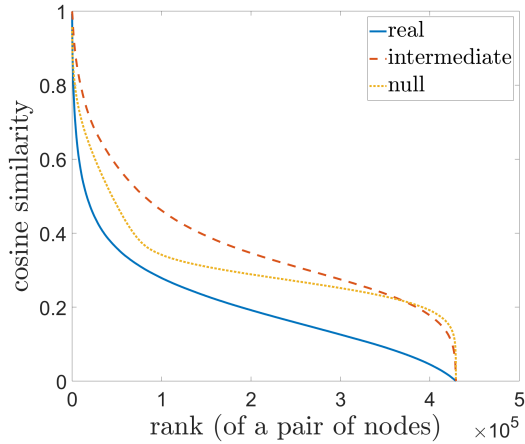


Fig. 10: Cosine similarity between pairs of nodes.

concentration of smaller cosine similarity scores (shifted on the left), and both 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 Porto move randomly) can be rejected.

V. RELATED WORK

A number of important works related to our research has already been cited throughout the manuscript. Here we further elaborate on other related work. In this survey paper [10],

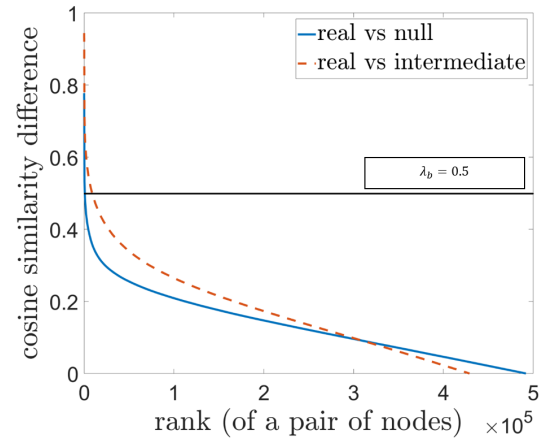


Fig. 11: Cosine similarity differences between pair of nodes.

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 unsupervised learning methods, based on network representation. A similar approach has been employed in [26] to learn low-rank embeddings of evolving networks. In [27] 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. [28] 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

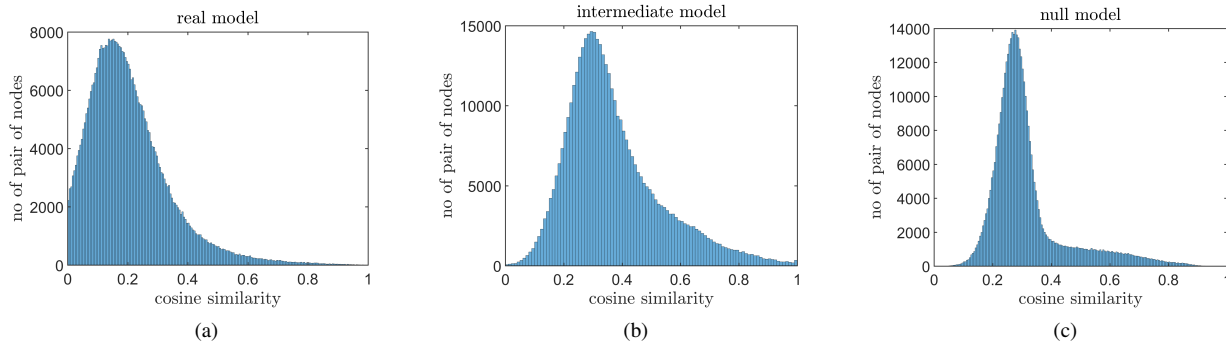


Fig. 12: Distribution of cosine similarity values between pair of nodes.

[29], authors discuss how to enrich trajectories with semantic information based on stop points in moving objects. Kumar et al. [30] 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. Urban planning, relieving traffic congestion, and effective location recommendations are important objectives worldwide and have received increasing attention in recent years. In this survey [31], the authors are introducing methods used for location prediction and recommendations and giving an insight into trajectory data pre-processing for different objectives. In [32] 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. Recent advances on trajectory data mining look on network dynamics of trajectories, such as mining group patterns of trajectories and mining the importance of a moving object in trajectory networks [33].

In [34] authors have developed a framework towards learning trajectory context by adapting the problem to an encoder-decoder framework. Our work differs in that we rely on graph embedding methods. In [17], 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 trajectories as walks and utilize them to learn embeddings in low-dimensional space. Ahmet et al. [35] introduced a data model PG-TRAJECTORY that is built on PostGIS, the spatial database extender of PostgreSQL. In [36]–[38] authors proposed methods to understand trajectories, mobility behavior of users and interesting locations by utilizing data-sets of users collected in Geolife project. 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 trajectories of moving objects.

VI. CONCLUSIONS

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.

Reproducibility: The source code and some execution examples are publicly available to encourage reproducibility of results. They can be accessed at the following website: https://github.com/saimmehmood/semantic_relationships

REFERENCES

- [1] M. van Kreveld and J. Luo, “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.
- [2] K. Toohy and M. Duckham, “Trajectory similarity measures,” *Sigspatial Special*, vol. 7, no. 1, pp. 43–50, 2015.
 - [3] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: a partition-and-group framework,” in *SIGMOD*, 2007, pp. 593–604.
 - [4] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, “A review of moving object trajectory clustering algorithms,” *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2017.
 - [5] A. Sawas, A. Abuolaim, M. Afifi, and M. Papagelis, “Trajectolizer: Interactive analysis and exploration of trajectory group dynamics,” in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2018, pp. 286–287.
 - [6] S. Dodge, R. Weibel, and E. Forootan, “Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects,” *Computers, Environment and Urban Systems*, vol. 33, no. 6, pp. 419–434, 2009.
 - [7] D. Guo, S. Liu, and H. Jin, “A graph-based approach to vehicle trajectory analysis,” *Journal of Location Based Services*, vol. 4, no. 3-4, pp. 183–199, 2010.
 - [8] K. Sita-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar, and A. S. Fotheringham, “Analysis of human mobility patterns from gps trajectories and contextual information,” *International Journal of Geographical Information Science*, vol. 30, no. 5, pp. 881–906, 2016.
 - [9] F. Zanlungo, T. Ikeda, and T. Kanda, “Potential for the dynamics of pedestrians in a socially interacting group,” *Physical Review E*, vol. 89, no. 1, p. 012811, 2014.
 - [10] Y. Zheng, “Trajectory data mining: an overview,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, p. 29, 2015.
 - [11] A. Sawas, A. Abuolaim, M. Afifi, and M. Papagelis, “A versatile computational framework for group pattern mining of pedestrian trajectories,” *GeoInformatica*, vol. 23, no. 4, pp. 501–531, 2019.
 - [12] T. Pechlivanoglou and M. Papagelis, “Fast and accurate mining of node importance in trajectory networks,” in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 781–790.
 - [13] G. Christoforidis, P. Kefalas, A. Papadopoulos, and Y. Manolopoulos, “Recommendation of points-of-interest using graph embeddings,” in *2018 IEEE 5th Int. Conf. on DSAA*. IEEE, 2018, pp. 31–40.
 - [14] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, “Learning graph-based poi embedding for location-based recommendation,” in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, 2016, pp. 15–24.
 - [15] B. Liu, Y. Fu, Z. Yao, and H. Xiong, “Learning geographical preferences for point-of-interest recommendation,” in *Proceedings of the 19th ACM SIGKDD*. ACM, 2013, pp. 1043–1051.
 - [16] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *KDD*. ACM, 2014, pp. 701–710.
 - [17] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *KDD*. ACM, 2016, pp. 855–864.
 - [18] R. Andersen, F. Chung, and K. Lang, “Local graph partitioning using pagerank vectors,” in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06)*. IEEE, 2006, pp. 475–486.
 - [19] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–369, 2007.
 - [20] P. D. Hoff, A. E. Raftery, and M. S. Handcock, “Latent space approaches to social network analysis,” *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1090–1098, 2002.
 - [21] F. Heidari and M. Papagelis, “Evolving network representation learning based on random walks,” *Applied Network Science*, vol. 5, no. 1, pp. 1–38, 2020.
 - [22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013, pp. 3111–3119.
 - [23] A. Anagnostopoulos, R. Kumar, and M. Mahdian, “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.
 - [24] M. Papagelis, V. Murdock, and R. van Zwol, “Individual behavior and social influence in online social systems,” in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, 2011, pp. 241–250.
 - [25] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
 - [26] F. Heidari and M. Papagelis, “Evonrl: Evolving network representation learning based on random walks,” in *International Conference on Complex Networks and their Applications*. Springer, 2018, pp. 457–469.
 - [27] H.-R. Wu, M.-Y. Yeh, and M.-S. Chen, “Profiling moving objects by dividing and clustering trajectories spatiotemporally,” *IEEE Trans. on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2615–2628, 2012.
 - [28] S. Shang, L. Chen, K. Zheng, C. S. Jensen, Z. Wei, and P. Kalnis, “Parallel trajectory-to-location join,” *IEEE TKDE*, 2018.
 - [29] B. Furlletti, P. Cintia, C. Renso, and L. Spinsanti, “Inferring human activities from gps tracks,” in *KDD*. ACM, 2013, p. 5.
 - [30] S. Kumar, X. Zhang, and J. Leskovec, “Predicting dynamic embedding trajectory in temporal interaction networks,” in *Proceedings of the 25th ACM SIGKDD*, 2019, pp. 1269–1278.
 - [31] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, “Location prediction on trajectory data: A review,” *BDMA*, vol. 1, no. 2, pp. 108–127, 2018.
 - [32] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng, “Planning bike lanes based on sharing-bikes’ trajectories,” in *Proceedings of the 23rd ACM SIGKDD*. ACM, 2017, pp. 1377–1386.
 - [33] A. Sawas, A. Abuolaim, M. Afifi, and M. Papagelis, “Tensor methods for group pattern discovery of pedestrian trajectories,” in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2018, pp. 76–85.
 - [34] F. Zhou, X. Yue, G. Trajcevski, T. Zhong, and K. Zhang, “Context-aware variational trajectory encoding and human mobility inference,” in *The World Wide Web Conference*, 2019, pp. 3469–3475.
 - [35] A. Kucuk, S. M. Hamdi, B. Aydin, M. A. Schuh, and R. A. Angryk, “Pg-trajectory: A postgresql/postgis based data model for spatiotemporal trajectories,” in *BDCloud-SocialCom-SustainCom*, 2016, pp. 81–88.
 - [36] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, “Understanding mobility based on gps data,” in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 312–321.
 - [37] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining interesting locations and travel sequences from gps trajectories,” in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.
 - [38] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, “Geolife: A collaborative social networking service among user, location and trajectory,” *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.