

ORIGINAL ARTICLE

MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings

Lucas May Petry^a, Camila Leite da Silva^a, Andrea Esuli^b, Chiara Renso^b and Vania Bogorny^a

^aPrograma de Pós-Graduação em Ciência da Computação (PPGCC) – Universidade Federal de Santa Catarina (UFSC) – Florianópolis, Brazil;

^bIstituto di Scienza e Tecnologie dell'Informazione (ISTI) – Consiglio Nazionale delle Ricerche (CNR) – Pisa, Italy

NOTICE: This is a **pre-peer reviewed** version of the article entitled **MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings**, accepted for publication in International Journal of Geographical Information Science (IJGIS), published by Taylor & Francis. Note that there may be significant changes from this version to the final published version.

ABSTRACT

The increasing popularity of Location-Based Social Networks (LSBNs) in the last few years is leading to the generation of large volumes of mobility data. In contrast to GPS-based trajectories, LSBN trajectories are more complex data, having besides space and time, several semantic textual dimensions, which may reveal interesting patterns about human mobility. These new semantically rich data, known as *multiple-aspect trajectories*, pose new challenges in trajectory classification, which is the problem that we address in this paper. Existing methods for trajectory classification cannot deal with the complexity of heterogeneous data dimensions, besides the sequential aspect that characterizes movement. In this paper we propose MARC, an approach based on attribute embedding and Recurrent Neural Networks (RNNs) for classifying multiple-aspect trajectories, that tackles all trajectory properties: space, time, semantics, and sequence. Experiments performed over four publicly available datasets show that MARC outperformed all competitors in all datasets.

KEYWORDS

trajectory classification; multiple-aspect trajectory; semantic trajectory classification; geohash embedding; recurrent neural network

1. Introduction

We are witnessing an explosion of big data generated on the internet and an increasing popularity of Location-Based Social Networks (LBSNs), such as Foursquare, Twitter, and Facebook. These networks collect not only information about visited places, feelings, and thoughts of their users, but their real movement, as sequences of *check-ins* or geotagged posts, that represent the sample points of a *trajectory*. We strongly believe that this new type of trajectory data is the challenge of the next generation data mining methods to be explored by large companies as Google and Facebook, for obtaining one of the most valuable information: the daily routine patterns and behaviors of every human being.

Semantically rich movement data are important for analyzing mobility patterns in a vast range of applications, from Point of Interest (POI) recommendation (Zhou et al. 2016, Feng et al. 2018), profiling taxi trip purposes (Chen et al. 2018), to trajectory classification (Ferrero et al. 2018, Zhou et al. 2018). In this work we focus on the trajectory classification problem, which consists of categorizing a moving object according to its trajectories (Lee et al. 2008). Examples are the transportation mode inference (e.g. car, bus, taxi) (Dabiri & Heaslip 2018, Etemad et al. 2018), determining a person profile (e.g. worker, student, retired) (de Alencar et al. 2015), the strength level of a hurricane (Lee et al. 2008, Ferrero et al. 2018) the user/owner of the trajectory (Gao et al. 2017, Zhou et al. 2018), among others.

Trajectory data are very complex because of the nature of their multiple dimensions. A raw trajectory, for instance, that is generated from a Global Positioning System (GPS) device, and that is the simplest type of movement data, is a sequence of spatio-temporal points in the form of (x, y, t) , where x and y represent the spatial position of the moving object at time instant t . Raw trajectories are more complex than time series, since they combine both time and space attributes, in which time is more than the temporal sequence.

Trajectory data extracted from LBSNs pose new challenges when compared to raw trajectories. While in raw trajectories a point has basically the space and time dimensions, in trajectories extracted from LBSNs a spatio-temporal point is enriched with several levels of heterogeneous semantic dimensions as the text of the social-media posts, the reviews of a venue, the humor/opinion of the moving object, etc. This new type of trajectory data is called *multiple-aspect trajectory* (Ferrero et al. 2016, Mello et al. 2019). Figure 1 shows an example of a multiple-aspect trajectory of a tourist in Paris. Besides the spatial and temporal dimensions, the trajectory is enriched with POI information (category, rating, and price), the means of transportation, the weather conditions, and the social-media posts of the tourist.

Differently from classical data mining problems, trajectory data cannot be summarized or normalized without losing information, and require a special treatment because of the complexity associated to the data. For instance, the spatial dimension is composed of two attributes, latitude and longitude, and they should be considered as a whole. Earlier methods for trajectory classification (Lee et al. 2008, Dodge et al. 2009, Patel 2013) have been developed for raw trajectories, where attributes and features are derived from the spatial and temporal dimensions, such as speed, acceleration, traveled distance, etc. In fact, existing works do not propose new classifiers, but propose different feature extraction and/or trajectory partition methods that better capture the trajectory discriminant parts.

A recent method, called MOVELETS, was proposed by Ferrero et al. (2018), supporting multiple attributes as space, time, and semantics. MOVELETS supports the

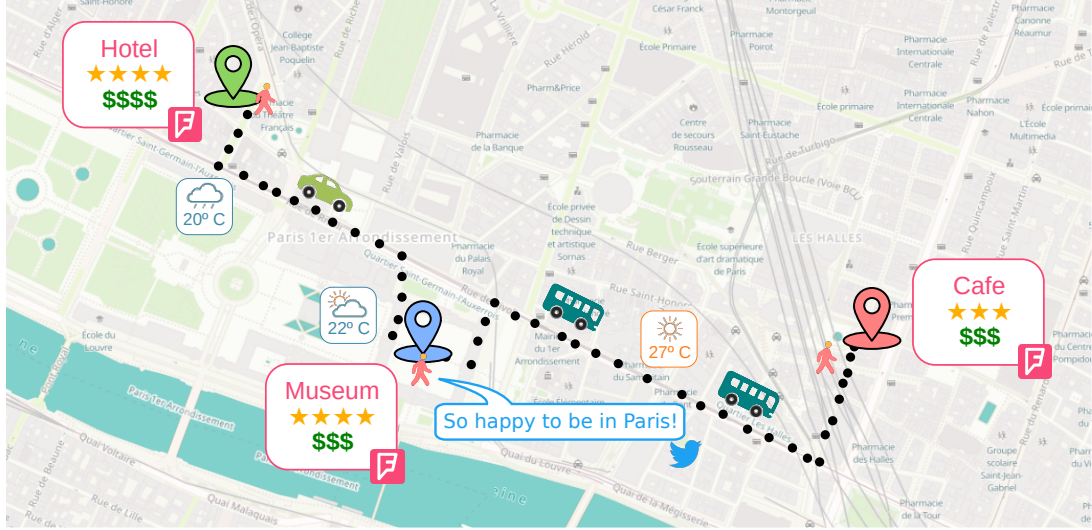


Figure 1.: The multiple-aspect trajectory of a tourist in Paris.

different types of data and the heterogeneous dimensions, and it outperformed most previous works for raw trajectories. Although it is robust for several dimensions, it only explores patterns of consecutive trajectory points, and does not leverage the aspects that better describe each class. Indeed, it is a time consuming method because it explores all possible subtrajectories of different size.

The works of Gao et al. (2017), Zhou et al. (2018), and Ferrero et al. (2019) have been developed for LBSN trajectory data. Differently from all previous works, Gao et al. (2017) and Zhou et al. (2018) use the idea of word embeddings, and embed POIs based on the concept of the distributional hypothesis, similarly to the idea proposed in Mikolov, Chen, Corrado & Dean (2013). An embedding is a numerical vector in an l -dimensional space \mathbb{R}^l , which can be a more meaningful representation of information for machine learning methods.

In Gao et al. (2017) and Zhou et al. (2018) a given POI is embedded based on the previous POIs and the next POI that the user visited. For instance, suppose a user has visited the POIs *Home*, *Restaurant*, and *Office*, while another user has visited the POIs *Home*, *Cafe*, and *Office*. The embeddings of *Restaurant* and *Cafe* will be similar, because they happened in the same context (after *Home* and before *Office*). However, in these works the embeddings are solely based on the POI sequence, so they neither explore the spatio-temporal dimensions nor the different semantic aspects that characterize multiple-aspect trajectories. We argue that the more trajectory aspects the classifier is able to treat, the more robust is the method and the better will be the classification accuracy.

In this paper we propose an approach for trajectory classification that is simpler than existing methods, more efficient, and that far outperforms the state of the art in four datasets. The method is a Recurrent Neural Network (RNN)-based approach for multiple-aspect trajectory classification via a multi-attribute embedding layer, which allows encoding the heterogeneous dimensions associated to each trajectory point. Because of the multiple and heterogeneous textual features that characterize human movement in social media data, more specifically due to the sparsity and high dimensionality of features, we believe that embeddings are the best technique to capture the semantic aspects of trajectories and that can deal together with the textual dimensions

and the spatial dimension represented with Geohash (Niemeyer 2008). In summary, we make the following contributions:

- We introduce a new classification method for trajectory data considering their multiple and heterogeneous dimensions that characterize current mobility data, called MARC. The MARC architecture exhibits similar or lower network complexity compared to state-of-the-art methods, while achieving better accuracy;
- We model trajectory points via a multi-attribute embedding layer, specifically including an approach for embedding the spatial dimension of trajectories using Geohash encoding (Niemeyer 2008), given that embeddings were designed for discrete data;
- We evaluate our method on four real-world LBSN datasets enriched with multiple semantic aspects. This allows us to show the robustness of our approach for classifying users based on their semantically rich trajectories. We compare our method with state-of-the-art approaches to show that MARC outperforms competitors in all datasets. We also highlight the fact that MARC performs well when information or trajectory attributes are missing.

The remainder of this paper is organized as follows: Section 2 presents related works and highlights their differences to our approach; Section 3 describes the proposed method; Section 4 presents the experimental evaluation, and Section 5 concludes the paper and outlines future work.

2. Related Work

Most existing works in trajectory classification are limited to raw spatio-temporal data. Such works focus on the extraction of features from trajectories, which are then given as input to traditional classifiers, such as Random Forest, Support Vector Machine, and Multilayer Perceptron (MLP) classifier. The features are extracted from the spatio-temporal points as, for instance, the average speed, acceleration, direction, etc, and are related either to the whole trajectory or to specific trajectory segments.

One of the first works for raw trajectory classification was the method TraClass (Lee et al. 2008), which is limited to the spatial dimension. Patel (2013) extended the work of Lee et al. (2008) to support the time dimension. Dodge et al. (2009) proposed a classification method that partitions trajectories based on deviation and sinuosity change, extracting attributes such as velocity, acceleration, turning angle, displacement and deviation change rate. The attributes are extracted from the partitions and the whole trajectory. Zheng et al. (2010) splits trajectories based on velocity and acceleration, further extracting statistics as the maximum velocity, maximum acceleration, deviation change, etc. Biljecki et al. (2013) propose a segmentation method for robust transportation mode classification. Recently, Xiao et al. (2017) extended the method proposed by Dodge et al. (2009) to extract a larger number of global and local attributes from trajectories.

Existing methods for raw trajectory classification were developed for dense GPS trajectories (e.g. points are regularly sampled every second), and are limited to numerical attributes inferred from space and time. In multiple-aspect trajectories extracted from LBSNs, the points can be more sparse (e.g. points can be sampled in hours or days), not being characterized by attributes such as speed, acceleration or direction change.

Ferrero et al. (2018) proposed MOVELETS, a method that supports spatial, temporal, and semantic attributes, which outperformed previous works for raw trajectory

classification. Ferrero explores the training set looking for relevant subtrajectories, which are the trajectory segments that better characterize the movement patterns of a given class. The presence or absence of the discovered MOVELETS in trajectories are then given as features to a traditional classifier. Although MOVELETS supports multiple attributes, it cannot automatically find those which better discriminate each class. Furthermore, MOVELETS cannot ignore noise in trajectory points, because it only explores patterns of consecutive trajectory points.

To the best of our knowledge, so far MOVELETS is the only method in the literature that can deal with multiple-aspect trajectories exploring space, time, semantics, and sequence. Novel methods were developed for LBSN trajectory classification (Gao et al. 2017, Zhou et al. 2018), but they are limited to a single trajectory attribute.

Gao et al. (2017) introduced the Trajectory-User Linking (TUL) problem, a special case of trajectory classification where the labels are the users, i.e., the person who generated each trajectory. Gao proposed Bi-TULER, a bidirectional RNN model for classifying LBSN trajectories. However, Gao analyzes only the POIs visited by the users, thus having trajectories represented as the sequence of places visited by a user. Bi-TULER learns continuous vector representations (embeddings) of the POIs, following the concept of the distributional hypothesis, similar to word embeddings proposed by Mikolov, Chen, Corrado & Dean (2013). The embeddings are used to train an RNN model for classifying trajectories, and, even though it is able to capture more complex patterns of mobility data than previous works, only the POI identifier is used for training the model. In Section 4 we show that using only one layer of semantics is insufficient for characterizing human movement. In fact, different levels of semantics provide a more accurate representation of human mobility.

More recently, Zhou extended Bi-TULER and proposed TULVAE (Zhou et al. 2018) to address the TUL problem. Similarly to Bi-TULER, POI embeddings are pre-learned and then fed to the model. Zhou employs a Variational Autoencoder (VAE) architecture to model trajectories and extracts an interpretable representation of POI dependencies present in trajectories. As Bi-TULER, TULVAE is solely based on the POIs visited by users, thus not supporting the spatial, temporal, and additional semantic attributes of trajectories. Furthermore, the model shows a relatively complex network architecture similar to sequence-to-sequence models.

Both Bi-TULER and TULVAE embed POIs similarly to the word embeddings described by Mikolov, Chen, Corrado & Dean (2013). Word embeddings alleviate data sparsity and provide a continuous representation of words (or POIs). However, the embedding of a POI is based on temporal context, regardless of class label, which can make POIs less discriminative in classification problems. The proposed method MARC, differently from Bi-TULER and TULVAE, is based on an end-to-end classification task, instead of pre-learning attribute embeddings similar to word embeddings.

3. Multiple-Aspect Trajectory Classification

In this section we introduce MARC: a novel method for Multiple-Aspect trajectory Classification. Figure 2 illustrates the three components of MARC: (1) trajectory encoding via a multi-attribute embedding layer; (2) a recurrent component for modelling the sequential factor present in trajectories; and (3) the classification component, which uses information from the previous components for assigning labels to trajectories.

MARC is a Recurrent Neural Network (RNN) that takes a trajectory as input and

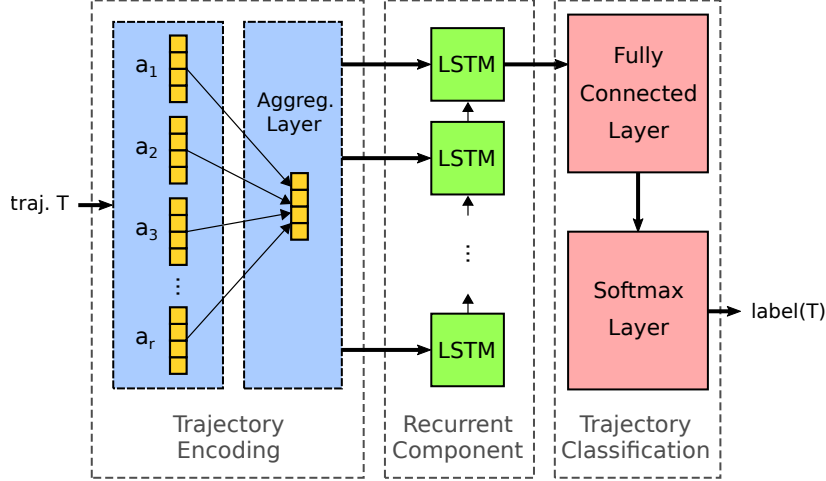


Figure 2.: Overview of the architecture of MARC.

outputs the corresponding label (class). RNNs are a special class of neural networks capable of processing sequences of inputs. As they demand higher computational power than normal feed-forward neural networks, their use in practical applications was only made possible in recent years. RNNs can be applied to a variety of applications, from speech recognition (Sak et al. 2014) to trajectory classification. Before going into the details of our method, we start defining *multiple-aspect trajectory* in Definition 3.1, based on the concept introduced by Ferrero et al. (2016).

Definition 3.1. *Multiple-aspect trajectory.* A multiple-aspect trajectory is a sequence of points $T = \langle p_1, p_2, \dots, p_n \rangle$, with $p_i = (x, y, t, \mathcal{A})$ being the i -th point of the trajectory at location (x, y) and timestamp t , described by the set $\mathcal{A} = \{a_1, a_2, \dots, a_r\}$ of r attributes.

A multiple-aspect trajectory is a trajectory enriched with semantic information, contributing to the characterization of the movement, such as the visited POIs, the goal of the visit, the possible activity, the weather conditions, the means of transportation, etc. Multiple-aspect trajectories are variable-length sequences of information, and the attributes of these trajectories are heterogeneous, have different natures, and distinct data types. In addition, as they may have several textual attributes, it may be harder for a classifier to measure the attribute similarity because of the high number of dimensions in the attribute space. Given their heterogeneity and sparsity, we use a multi-attribute embedding layer in order to encode these multiple attributes, which is detailed in the following section.

3.1. Trajectory Encoding

The first component of our method is responsible for encoding trajectory attributes. Since trajectory attributes can have a variety of formats, we use an embedding layer for uniformly encoding them. In neural networks, embedding layers map attributes to an embedding space, so that they may be fed to the subsequent layers of the network.

Embeddings are numerical vector representations in a continuous l -dimensional space \mathbb{R}^l , created according to a model (e.g. vector space models for modeling textual data). Attributes are usually embedded in order to reduce the dimensionality of

the underlying space, so that the similarity of attributes can be better measured; or simply to map discrete attributes into equivalent continuous meaningful representations. For instance, let us consider the POIs Park, Restaurant, and Cafe. If we compare them with a string similarity function such as the Edit Distance (ED) (Wagner & Fischer 1974), Cafe and Park will have a higher similarity than Cafe and Restaurant, which is not realistic from the semantic point of view. Hence, embedding methods are used to create similarity-based representations for textual attributes. As the number of attributes can grow fast in multiple-aspect trajectories, it may be unfeasible and hard to define similarity measures for every attribute, as proposed in Ferrero et al. (2018) and Furtado et al. (2016), for instance.

Mikolov, Chen, Corrado & Dean (2013) proposed a neural network model to embed words based on the context they appear in the text (words that come before and after a given word). While Bi-TULER (Gao et al. 2017) and TULVAE (Zhou et al. 2018) embed a POI considering the previous and the next POI, in our method we not only embed POIs but all trajectory attributes, which we describe later. We formally define the embedding of a single attribute as follows.

Definition 3.2. *Attribute embedding.* Given a trajectory $T = \langle p_1, p_2, \dots, p_n \rangle$ and a set $\mathcal{A} = \{a_1, a_2, \dots, a_r\}$ of attributes describing p_i , the embedding of attribute a_k is given by $e(a_k) = \text{encoding}(a_k) \mathbb{W}_{a_k}$, where \mathbb{W}_{a_k} is the embedding matrix of attribute a_k .

\mathbb{W}_{a_k} is a matrix with $|a_k| \times l$ dimensions, where $|a_k|$ is the number of values that attribute a_k may assume. For instance, if we have 30 different POIs in a dataset and decide to embed them in the space \mathbb{R}^{50} , \mathbb{W}_{POI} will have 30×50 dimensions. For nominal and numeric attributes, $\text{encoding}(a_k)$ is the one-hot encoding of attribute a_k , so that embeddings are properly selected from the embedding matrix. The one-hot encoding of attribute a_k is a $|a_k|$ -dimensional vector of zeros, with a 1 in the position corresponding to the value of the attribute. In other words, given an ordering of the values of a_k , the $\text{encoding}(a_k)$ is built in a way so that when multiplied by \mathbb{W}_{a_k} , the i -th row of \mathbb{W}_{a_k} is the embedded representation of the i -th value of a_k .

The spatial dimension of trajectories is usually modeled with two attributes, latitude and longitude, and these attributes are meaningless if considered separately. Therefore, we encode them with the Geohash algorithm (Niemeyer 2008), as it is necessary to merge them into a single meaningful attribute in order to be used with embeddings.

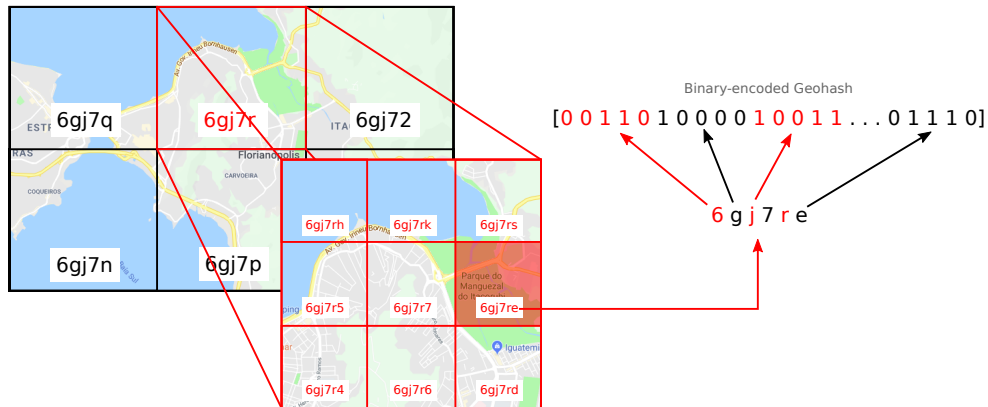


Figure 3.: Geohash encoding toy example.

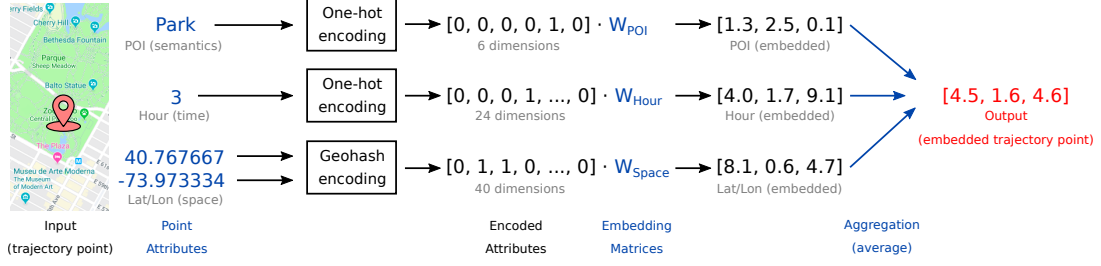


Figure 4.: Trajectory point embedding example.

Figure 3 illustrates the Geohash algorithm. It successively divides the space into buckets of grid shape, encoding a spatial location (latitude and longitude coordinates) as a Base32 character string. Two locations with a common Geohash prefix are spatially close to each other. We further extract the binary representation of Geohash, and use it as the encoding of the spatial dimension of trajectory points. Since Base32 is used for building the Geohash encoding, each character is mapped to a 5-digit binary string ($2^5 = 32$). For instance, considering 32 characters from 0 to 9 and A to V, 0 is mapped to 00000, 1 is mapped to 00001, 2 to 00010, and so on up to V, which is mapped to 11111.

Figure 4 exemplifies the embedding process of trajectory points with semantic, temporal, and spatial attributes. The input is a trajectory point with the dimensions space (40.767667, -73.97334), time (3), and semantics (Park). The attributes are first one-hot encoded, except for spatial attributes, that are encoded with Geohash. In this example the dataset has only 6 different POIs, so the one-hot encoded POI vector has 6 dimensions, and Park corresponds to the 5-th element in the vector. Similarly, as there are 24 different hours, the encoded hour is a 24-dimensional vector and the 4-th element corresponds to the hour 3 (hours encoded from 0 to 23). On the other hand, the space dimension is encoded with Geohash. The size of the encoded spatial location depends on the precision chosen for the Geohash algorithm, i.e., how many buckets we have in the grid (see Figure 3). Subsequently, the encoded attributes are multiplied by their respective embedding matrices to extract their corresponding embedded representations.

In order to encode trajectory points with multiple attributes, we apply an aggregation function to the embedded point attributes. Attributes must be aggregated so that trajectory points can be fed to the recurrent component of the network. We may combine attributes similarly to how words are aggregated in Le & Mikolov (2014), via element-wise average or concatenation. In Figure 4, we aggregate the embedded attributes by averaging each dimension individually. This implies that if we aggregate attributes by sum or average, all embeddings must have the same size. If we opt for concatenation, then the embedded attributes may have different sizes, and the size of the final encoded trajectory point will be the sum of the embedding sizes. In Section 4 we present different results considering the sum, average, and concatenation of attributes.

3.2. Recurrent Component

To properly assess trajectory data, we use an RNN with Long Short-Term Memory (LSTM) units (Hochreiter & Schmidhuber 1997), which is the state of the art for sequence processing in neural networks. RNNs are able to represent more complex

patterns than shallow networks, and can deal with variable-length sequences of information.

After encoding trajectory points, trajectories are fed to the recurrent module. LSTM cells capture patterns in sequences over variable-length time intervals, regulating how much information is remembered via their input, output, and forget gates. Compared to MOVELETS (Ferrero et al. 2018), for instance, this can be an advantage for modeling trajectory patterns. MOVELETS only captures sequential patterns of consecutive trajectory points, while, for instance, LSTM cells may learn patterns that include the very first points and the last points of a trajectory. This means that LSTM units can model the relationships between different trajectory points and their attributes, even if they are far away.

3.3. *Trajectory Classification*

The last component of our method uses the information provided by the recurrent component for inferring the trajectory label. The output of the recurrent module is fed into a fully-connected layer and subsequently a softmax function is applied. The goal of the last fully-connected layer is to map the learned knowledge to the corresponding label. Then, a softmax function is applied to emphasize the differences between labels, further outputting the probability distribution for all possible data labels. As for some classification problems the number of labels can be particularly high, such as the TUL problem introduced by Gao et al. (2017), negative sampling (Mikolov, Sutskever, Chen, Corrado & Dean 2013) can be employed to alleviate the cost of softmax computation.

3.4. *Training and Optimization*

For training the proposed model, our goal is to minimize the categorical cross entropy loss, as given by the following equation

$$-\frac{1}{\mathcal{T}_{train}} \sum_{i=1}^{\mathcal{T}_{train}} \sum_{l=1}^{\mathcal{L}} 1_{T_i \in \mathcal{L}_k} \log p[T_i \in \mathcal{L}_k] \quad (1)$$

where \mathcal{T}_{train} is the number of trajectories in the training set and \mathcal{L} is the number of labels. In other words, we want to maximize the probability of our model to correctly predict the label of each trajectory T_i .

To avoid overfitting our model to the training data, which is a problem inherent to deep neural networks such as RNNs, we use dropout (Srivastava et al. 2014) and regularization techniques. Dropout layers are applied throughout the model, so units are randomly dropped during the training process. In addition, the weights and biases of the LSTM units are regularized using L1 regularization.

As in the works of Gao et al. (2017) and Zhou et al. (2018), the embedding layers of the input attributes could be initialized by training embeddings in a separate model, similar to learning word embeddings (Mikolov, Chen, Corrado & Dean 2013). Hence, attribute values that appear in similar contexts in trajectories will have similar embeddings. Although this method is good for capturing attributes similarity, these embeddings may substantially spoil the performance of a classifier, making it harder to discriminate between different classes (labels).

In the following section we present an experimental evaluation, showing the robustness of our method in relation to state-of-the-art approaches.

4. Experimental Evaluation

We evaluate our approach over four real-world trajectory datasets extracted from the Foursquare, Brightkite, and Gowalla LSBNs. These datasets have been widely used in several works on both trajectory classification (Gao et al. 2017, Zhou et al. 2018) and next POI prediction (Zhao et al. 2017, Feng et al. 2018).

We compare our work to the state-of-the-art methods that can handle trajectories that have other dimensions than space and time. These works are Bi-TULER (Gao et al. 2017) and TULVAE (Zhou et al. 2018), developed for LBSN trajectories, and MOVELETS (Ferrero et al. 2018), that outperformed previous methods developed for raw trajectories. We evaluate existing works using a similar approach to the Trajectory-User Linking (TUL) problem described in Gao et al. (2017), in which the classification task is to predict the corresponding user who generated a given trajectory. For reproducibility purposes, we made the source code of MARC available on GitHub¹.

In the next few sections, we describe the datasets, the metrics used to evaluate the results, the experimental setup, and the achieved results.

4.1. Datasets

We run the experiments over four datasets extracted from the Foursquare, Brightkite, and Gowalla LSBNs: (i) a dataset of Foursquare check-ins in New York, USA, collected between April 2012 and February 2013 (Yang et al. 2015); (ii) a dataset of Foursquare check-ins in several cities around the world, collected between April 2012 and September 2013 (Yang et al. 2016); (iii) a dataset of user check-ins around the world on Brightkite, collected between April 2008 and October 2010 (Cho et al. 2011); and (iv) a dataset of user check-ins around the world on the Gowalla LBSN, collected between February 2009 and October 2010 (Cho et al. 2011).

Tables 1 and 2 describe the attributes of trajectory points in each dataset. In order to show that our approach can handle many dimensions and that additional information contribute to the classification of trajectories, we enrich the original dataset with more information. For the Foursquare datasets, we enriched check-in data with venue information (e.g. price tier and rating) collected from the Foursquare API². In addition, weather information was collected from the Weather Wunderground API³ and added to each check-in in the Foursquare NYC dataset. The attribute *User ID* is the label of each trajectory in the datasets.

In order to ensure variability and consistency in the evaluation, we applied a few transformations to the datasets. For the Foursquare datasets we removed noisy check-ins belonging to broad categories, such as roads and neighborhoods, because each venue has a unique geographic location. We also removed duplicated check-ins considering a 10-minute threshold. For all datasets, we created weekly trajectories from each user check-in, and we selected only trajectories with at least 10 check-ins, as well as users who have at least 10 weekly trajectories. For the Gowalla and Brightkite datasets we randomly selected 300 users. Table 3 shows the statistics of the curated datasets. As may be observed, the evaluated datasets are heterogeneous, with different sizes, number of classes, and trajectories.

¹<https://github.com/bigdata-ufsc/petry-2019-marc>

²<https://developer.foursquare.com/>

³<https://www.wunderground.com/weather/api/>

Table 1.: Attributes description of Foursquare NYC and Foursquare Global.

Attribute	Type	Range/example	N.
POI	Nominal	{15703, 21580, ...}	{13,848, 131,155}
Space	Composite	{(38.7423, -90.3658), ...}	-
Category	Nominal	{Arts & Entertainment, College & University, ...}	10
Price tier	Numeric	{-999, 1, 2, 3, 4}	5
Rating	Numeric	{-999} \cup [4.0, 10.0]	62
Hour	Numeric	[0, 23]	24
Weekday	Nominal	{Monday, Tuesday, ..., Sunday}	7
Weather ¹	Nominal	{Clear, Clouds, Rain, ...}	6
User ID	Nominal	{1, 2, 3, ...}	{193, 498}

¹ Available only for the Foursquare NYC dataset.

Table 2.: Attributes description of Brightkite and Gowalla.

Attribute	Type	Range/example	N.
POI	Nominal	{19959, 74884, ...}	{4,913, 24,374}
Space	Composite	{(38.7423, -90.3658), ...}	-
Hour	Numeric	[0, 23]	24
Weekday	Nominal	{Monday, Tuesday, ..., Sunday}	7
User ID	Nominal	{142, 143, ...}	300

4.2. Metrics

For each dataset we report ACC@K, Macro-P (precision), Macro-R (recall), and Macro-F1, which are commonly used metrics in classification and information retrieval. ACC@K shows how well each technique correctly estimates the probability of the correct trajectory labels among the K most probable labels. We compute ACC@K as

$$\text{ACC@K} = \frac{1}{\mathcal{T}_{test}} \sum_{i=1}^{\mathcal{T}_{test}} \begin{cases} 1, & \text{if label}(T_i) \in \mathcal{L}_k(T_i) \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where \mathcal{T}_{test} is the number of trajectories in the test set, $\text{label}(T_i)$ is the true label of trajectory T_i , and $\mathcal{L}_k(T_i)$ is the set of k labels with the highest probabilities predicted for trajectory T_i . Macro-P and Macro-R are the mean precision and recall among all classes, respectively. Macro-F1 is the harmonic mean of Macro-P and Macro-R, averaged across all classes, computed as

$$\text{Macro-F1} = \frac{2 \times \text{Macro-P} \times \text{Macro-R}}{\text{Macro-P} + \text{Macro-R}} \quad (3)$$

4.3. Experimental Setup

On all the datasets we run the experiments performing a stratified holdout evaluation, with 2/3 of the data for training and 1/3 for validation. In order to show the robustness

Table 3.: Summary of the datasets (averages reported in the format $\text{AVG} \pm \text{SD}$).

Dataset	# of points	# of trajectories	# of labels	Avg. traj. length	Avg. # of traj. per class
Brightkite	130,494	7,911	300	16.50 ± 7.07	26.37 ± 16.37
Foursquare Global	663,020	20,911	498	31.71 ± 22.05	41.99 ± 13.04
Foursquare NYC	66,962	3,079	193	21.75 ± 14.46	15.95 ± 6.33
Gowalla	98,158	5,329	300	18.42 ± 8.05	17.76 ± 8.20

of our work, we run an additional experiment on a modified version of the Foursquare datasets in which the POI identifier is generalized to POI category and the spatial dimension is removed. This is motivated by the fact that human movement exhibits high spatio-temporal regularity, so users tend to visit a few POIs regularly (Gonzalez et al. 2008). Therefore, the POIs visited by a user, as well as the spatial dimension, are highly discriminative information, and, because of privacy concerns (Seidl et al. 2016), in some situations the exact locations visited by users may not be publicly available. Hence, we consider this further experiment to be representative in a realistic scenario, which is harder than the one typically faced in previous literature.

We use 100-dimensional embeddings and 100 LSTM units for MARC and we run three variants of our model: MARC-S, MARC-A, and MARC-C, which use sum, average, and concatenation for attribute aggregation, respectively. We also run a variant of our method, named MARC (Geohash), with only the space dimension, so that we can validate the use of the Geohash representation in the model.

For Bi-TULER and TULVAE we use the same settings reported in the respective papers. We embed POIs into 250-dimensional vectors and use 300 units for the classifier RNNs. We use 512 units for the encoder-decoder RNN and 100 units for the latent variable z in TULVAE. For MOVELETS, we experimented several attributes and their combinations and report the best results. For the experiment where the POI identifier is removed we consider all trajectory attributes. Additionally, as we run MOVELETS for multiple-aspect trajectories, we use binary distance for nominal features and euclidean distance for numeric, temporal, and spatial dimensions. We evaluated MOVELETS with a single-layer MLP classifier with 100 units, Decision Trees and Random Forest, and only the best results (achieved with MLP) are reported. For all networks we use a dropout rate of 0.5, batch size of 64, and we minimize the categorical cross entropy loss using the Adam optimizer and a learning rate of 10^{-3} .

In the following sections we present the experimental results and discussion. In Section 4.4 we show the classification results on the original datasets, and in Section 4.5 we discuss the results on the modified version of the Foursquare datasets.

4.4. Classification Results and Discussion

Table 4 shows the classification results for the proposed method and compared to existing techniques over the four datasets. For each metric, the best result is highlighted in bold and the second best result is underlined. The results show that MARC systematically outperforms existing methods on all datasets. MOVELETS is the second best method on Foursquare NYC, Brightkite (tied with Bi-TULER), and Gowalla, which used only the POI identifiers to classify trajectories. Such results show that, indeed,

Table 4.: Classification results on stratified holdout evaluation, considering the datasets with all attributes.

Dataset	Metric	Bi-TULER (Gao et al. 2017)	MOVELETS (Ferrero et al. 2018)	TULVAE (Zhou et al. 2018)	MARC-S	MARC-A	MARC-C	MARC (Geohash)
Foursquare NYC	ACC@1	48.20	97.66	54.33	97.57	<u>98.05</u>	98.64	93.77
	ACC@5	67.38	<u>98.93</u>	73.81	<u>98.93</u>	<u>98.93</u>	99.03	97.86
	Macro-P	43.48	96.85	48.66	<u>97.94</u>	97.77	98.34	94.33
	Macro-R	41.88	96.67	47.32	96.94	<u>97.50</u>	98.24	92.56
	Macro-F1	40.56	96.45	46.54	97.11	<u>97.37</u>	98.20	92.75
Foursquare Global	ACC@1	80.58	76.66 ¹	80.67	<u>98.78</u>	98.65	99.07	96.67
	ACC@5	90.23	80.38 ¹	90.48	99.33	99.20	<u>99.31</u>	98.87
	Macro-P	81.69	75.88 ¹	81.01	<u>98.78</u>	98.64	99.07	96.67
	Macro-R	77.57	68.13 ¹	77.60	<u>98.63</u>	98.53	98.98	96.24
	Macro-F1	78.31	70.27 ¹	78.07	<u>98.64</u>	98.52	98.97	96.28
Brightkite	ACC@1	90.64	90.55	88.41	<u>96.35</u>	96.01	96.85	94.92
	ACC@5	95.55	93.79	92.15	<u>98.62</u>	<u>98.62</u>	98.99	97.61
	Macro-P	88.98	93.53	85.00	<u>95.97</u>	95.85	96.76	93.90
	Macro-R	88.23	87.92	83.86	<u>95.43</u>	95.00	96.21	93.19
	Macro-F1	87.92	89.41	83.63	<u>95.15</u>	94.86	96.17	93.04
Gowalla	ACC@1	66.15	91.44	67.94	92.36	<u>93.46</u>	94.33	88.31
	ACC@5	78.36	94.04	78.76	96.01	<u>97.16</u>	97.57	95.31
	Macro-P	67.44	93.24	69.19	92.72	<u>94.11</u>	94.77	88.93
	Macro-R	63.21	89.27	64.80	91.25	<u>92.63</u>	93.28	86.90
	Macro-F1	63.26	90.25	64.91	91.13	<u>92.43</u>	93.28	86.81

¹ We could not run MOVELETS without filtering the set of discovered MOVELETS, because too many are generated.

the POIs visited by users are highly discriminative data for trajectory classification. Moreover, the high space complexity of MOVELETS is shown by the fact that we were not able to run MOVELETS without filtering the set of discovered patterns on Foursquare Global, the largest dataset evaluated.

Bi-TULER and TULVAE perform poorly on Foursquare NYC (accuracies of 48.20 and 54.33, respectively), but significantly better on Foursquare Global (accuracies of 80.58 and 80.67, respectively), Brightkite (accuracies of 90.64 and 88.41, respectively), and Gowalla (accuracies of 66.15 and 67.94, respectively). We conjecture that these results are related to (1) the larger size of these datasets and (2) the geographic distribution of check-ins. Check-ins in the Foursquare Global, Brightkite, and Gowalla datasets are distributed around the globe, so the data is inherently more discriminative than in the Foursquare NYC dataset.

Considering the three variations of our method, we observe that aggregating attributes via concatenation (MARC-C) yielded the best results for the majority of the metrics. However, summing (MARC-S) and averaging (MARC-A) attributes also gave great results, on average no more than 1% above or below concatenation scores. As stated in Section 3, MARC-C uses embedded representations and weight matrices r times larger than MARC-S and MARC-A (considering the same embedding dimension for all attributes). For applications with memory/storage constraints, MARC-S and MARC-A are the best alternatives. Considering only the spatial dimension (MARC (Geohash)), we observe that MARC was able to achieve competitive results with the compared approaches. The high scores achieved by MARC (Geohash) show that the Geohash representation can successfully preserve the discriminative power of the space dimension in neural networks.

Figure 5 shows the convergence of ACC@1 for all methods on all datasets. All variants of MARC show fast convergence in comparison to existing deep learning methods, Bi-TULER and TULVAE. Both Bi-TULER and TULVAE pre-learn embeddings for POIs in an unsupervised manner, which we claim to be one of the underlying factors for their worse performance. MOVELETS exhibits fast convergence as well, but it is important to highlight that it uses a shallow neural network and there is an extensive

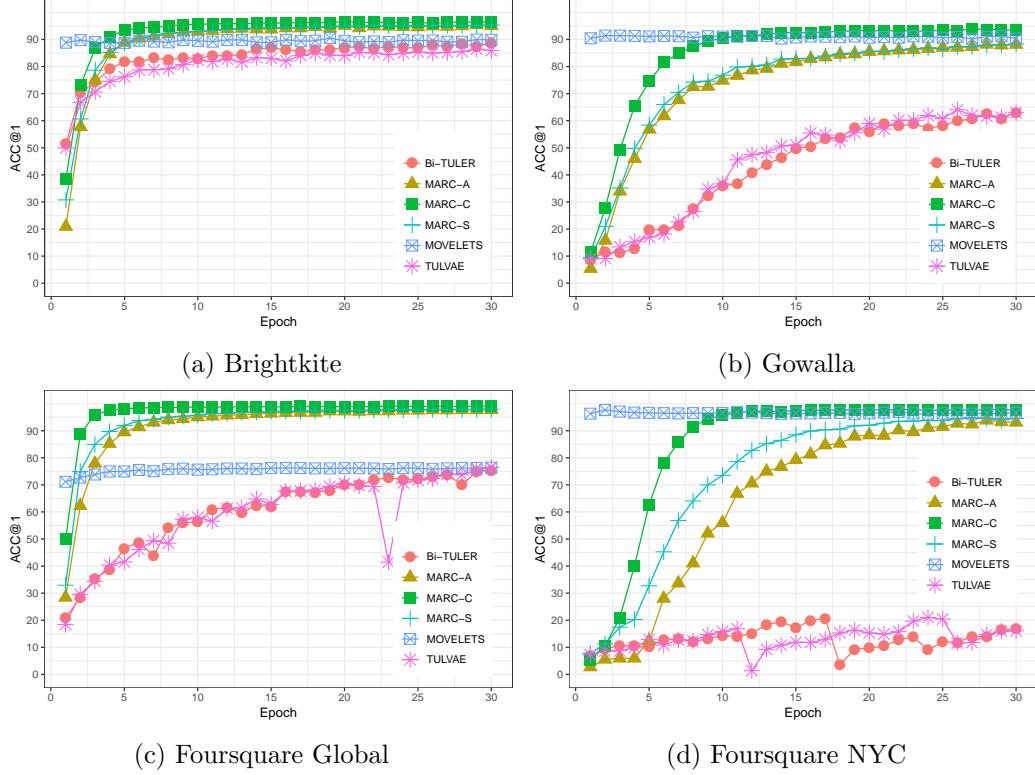


Figure 5.: ACC@1 of compared approaches, considering the datasets with all attributes.

feature extraction process before the classification task is performed.

4.5. Classification Results for Generalized POIs and Without the Spatial Dimension

Table 5 presents the results for the experiment where the POI identifier was generalized to POI category and the spatial dimension removed. We highlight that this experiment shows a more difficult yet realistic scenario, motivated by privacy concerns about the users information. Similarly to the previous experiment, for each metric the best result is highlighted in bold and the second best result is underlined. We observe that for all methods the accuracy decreased significantly in comparison to the previous experiment in which the detailed information of the POI identifier and the spatial dimension was considered. However, among all approaches, MARC still keeps a significantly higher accuracy when compared to state-of-the-art.

We explain the poor performance of previous works with two major aspects of this experiment. First, as we observed in the previous experiment, the specific POI identifiers play a big role in discriminating user trajectories. Bi-TULER and TULVAE consider only the POI attribute, which was replaced by the POI category for this experiment. Since they do not consider multiple trajectory attributes, their performance decreased substantially, achieving 33.50 and 32.81 accuracy on Foursquare NYC, and 34.70 and 34.40 on Foursquare Global, respectively. Second, classifiers must be able to leverage between the remaining attributes in order to correctly classify trajectories.

Although the MOVELETS technique supports multiple attributes, it always con-

Table 5.: Classification results on stratified holdout evaluation, considering the datasets without the POI identifiers.

Dataset	Metric	Bi-TULER (Gao et al. 2017)	MOVELETS (Ferrero et al. 2018)	TULVAE (Zhou et al. 2018)	MARC-S	MARC-A	MARC-C
Foursquare NYC	ACC@1	33.50	41.58	32.81	<u>74.10</u>	70.59	74.87
	ACC@5	60.76	67.96	61.15	88.32	87.83	<u>88.12</u>
	Macro-P	29.58	38.76	29.55	<u>73.16</u>	70.83	75.19
	Macro-R	29.70	36.93	29.52	<u>71.28</u>	67.34	72.23
	Macro-F1	28.29	35.11	28.14	<u>70.21</u>	66.59	71.22
Foursquare Global	ACC@1	34.70	51.88	34.40	81.97	81.47	<u>81.55</u>
	ACC@5	60.46	73.74	62.01	<u>92.73</u>	92.74	92.28
	Macro-P	34.22	54.64	33.64	<u>82.47</u>	82.55	82.20
	Macro-R	33.43	49.96	32.96	80.53	80.16	<u>80.34</u>
	Macro-F1	32.41	49.19	31.58	80.63	80.40	<u>80.46</u>

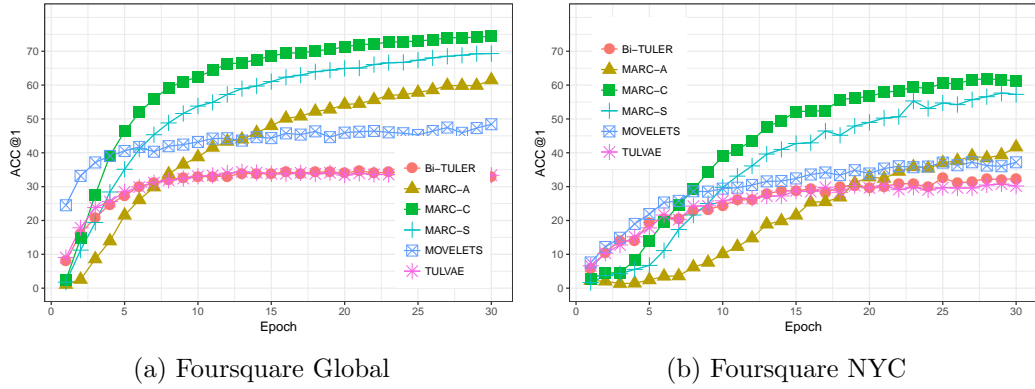


Figure 6.: ACC@1 of compared approaches, considering the datasets without the POI identifiers.

siders all attributes when looking for trajectory patterns, and the less discriminant dimensions will then add noise to the pattern, as MOVELETS is not able to find the best dimensions for classification problems. MOVELETS achieved 41.58 and 51.88 accuracies on Foursquare NYC and Global, respectively, and performed better than Bi-TULER and TULVAE for all evaluated metrics on both datasets. However, by embedding multiple attributes and modelling complex sequential patterns with MARC, we were able to achieve much higher accuracy and F1 scores than existing approaches (between 70% and 92%).

Figure 6 shows the convergence of ACC@1 for all methods in this new scenario. MARC achieves accuracies lower than the ones of Bi-TULER and TULVAE for the first few epochs, because while they use pre-trained embeddings, we train an end-to-end task. Afterwards, our models continue to learn whereas both Bi-TULER and TULVAE converge to an accuracy of about 33%. MOVELETS also exhibits much slower convergence compared to the previous experiment, which suggests that the discovered patterns are not as discriminating as they were before. In summary, the results of the experiments show that MARC is more robust than existing approaches, and good classification accuracies can be achieved after only a few epochs of training.

5. Conclusions

In this paper we presented a new method, named MARC, for classifying multiple-aspect trajectories. Our method focuses on the different spatial, temporal, and semantic attributes that characterize multiple-aspect trajectories, and a multi-attribute embedding layer is used to encode these heterogeneous dimensions. We leave to the neural network the task of learning abstract features and sequential patterns that are present in trajectory data. We designed an architecture with similar or lower network complexity compared to existing works, yet it achieved significantly higher levels of accuracy than other state-of-the-art approaches.

As future work, we will investigate the use of attention mechanisms for modelling trajectory patterns, as it has shown to be a good approach for trajectory POI prediction (Feng et al. 2018). Furthermore, we want to consider supervised approaches for embedding attributes or trajectories, such as the use of the distance-based histogram loss (Ustinova & Lempitsky 2016) and prototypical networks (Snell et al. 2017), because we believe that trajectory attributes lose their discriminative power when embedded through an unsupervised manner as in previous works.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and the European Union’s Horizon 2020 research and innovation programme under GA N. 777695 (MASTER⁴). The views and opinions expressed in this paper are the sole responsibility of the author and do not necessarily reflect the views of the European Commission.

References

- Biljecki, F., Ledoux, H. & van Oosterom, P. (2013), ‘Transportation mode-based segmentation and classification of movement trajectories’, *International Journal of Geographical Information Science* **27**(2), 385–407.
URL: <https://doi.org/10.1080/13658816.2012.692791>
- Chen, C., Liao, C., Xie, X., Wang, Y. & Zhao, J. (2018), ‘Trip2vec: a deep embedding approach for clustering and profiling taxi trip purposes’, *Personal and Ubiquitous Computing* pp. 1–14.
- Cho, E., Myers, S. A. & Leskovec, J. (2011), Friendship and mobility: user movement in location-based social networks, in ‘Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining’, ACM, pp. 1082–1090.
- Dabiri, S. & Heaslip, K. (2018), ‘Inferring transportation modes from gps trajectories using a convolutional neural network’, *Transportation Research Part C: Emerging Technologies* **86**, 360–371.
- de Alencar, L. A., Alvares, L. O., Renso, C., Raffaetà, A. & Bogorny, V. (2015), A rule-based method for discovering trajectory profiles, in ‘The 27th International Conference on Software Engineering and Knowledge Engineering, SEKE 2015, Wyndham Pittsburgh University Center, Pittsburgh, PA, USA, July 6-8, 2015’, pp. 244–249.
URL: <https://doi.org/10.18293/SEKE2015-143>

⁴<http://www.master-project-h2020.eu/>

- Dodge, S., Weibel, R. & Forootan, E. (2009), ‘Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects’, *Computers, Environment and Urban Systems* **33**(6), 419–434.
- Etemad, M., Soares Júnior, A. & Matwin, S. (2018), Predicting transportation modes of gps trajectories using feature engineering and noise removal, *in* ‘Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada, May 8–11, 2018, Proceedings 31’, Springer, pp. 259–264.
- Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A. & Jin, D. (2018), Deepmove: Predicting human mobility with attentional recurrent networks, *in* ‘Proceedings of the 2018 World Wide Web Conference on World Wide Web’, International World Wide Web Conferences Steering Committee, pp. 1459–1468.
- Ferrero, C. A., Alvares, L. O. & Bogorny, V. (2016), Multiple aspect trajectory data analysis: research challenges and opportunities, *in* ‘XVII Brazilian Symposium on Geoinformatics - GeoInfo 2016, Campos do Jordão, SP, Brazil, November 27-30, 2016.’, pp. 56–67.
URL: <http://urlib.net/8JMKD3MGPDW34P/3NDC42E>
- Ferrero, C. A., Alvares, L. O., Zalewski, W. & Bogorny, V. (2018), Movelets: Exploring relevant subtrajectories for robust trajectory classification, *in* ‘Proceedings of the 33rd ACM/SIGAPP Symposium on Applied Computing, Pau, France’, pp. 9–13.
- Ferrero, C. A., Petry, L. M., Alvares, L. O., Zalewski, W. & Bogorny, V. (2019), ‘Discovering heterogeneous subsequences for trajectory classification’, *arXiv preprint arXiv:1903.07722*.
- Furtado, A. S., Kopanaki, D., Alvares, L. O. & Bogorny, V. (2016), ‘Multidimensional similarity measuring for semantic trajectories’, *Transactions in GIS* **20**(2), 280–298.
- Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X. & Zhang, F. (2017), Identifying human mobility via trajectory embeddings, *in* ‘Proceedings of the 26th International Joint Conference on Artificial Intelligence’, AAAI Press, pp. 1689–1695.
- Gonzalez, M. C., Hidalgo, C. A. & Barabasi, A.-L. (2008), ‘Understanding individual human mobility patterns’, *nature* **453**(7196), 779.
- Hochreiter, S. & Schmidhuber, J. (1997), ‘Long short-term memory’, *Neural computation* **9**(8), 1735–1780.
- Le, Q. & Mikolov, T. (2014), Distributed representations of sentences and documents, *in* ‘International Conference on Machine Learning’, pp. 1188–1196.
- Lee, J.-G., Han, J., Li, X. & Gonzalez, H. (2008), ‘Trajclass: trajectory classification using hierarchical region-based and trajectory-based clustering’, *Proceedings of the VLDB Endowment* **1**(1), 1081–1094.
- Mello, R. d. S., Bogorny, V., Alvares, L. O., Santana, L. H. Z., Ferrero, C. A., Frozza, A. A., Schreiner, G. A. & Renso, C. (2019), ‘Master: A multiple aspect view on trajectories’, *Transactions in GIS* **23**(4).
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12526>
- Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), ‘Efficient estimation of word representations in vector space’, *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* ‘Advances in neural information processing systems’, pp. 3111–3119.
- Niemeyer, G. (2008), ‘Geohash’.
URL: <https://en.wikipedia.org/wiki/Geohash>
- Patel, D. (2013), ‘Incorporating duration and region association information in trajectory classification’, *Journal of Location Based Services* **7**(4), 246–271.

- URL:** <https://doi.org/10.1080/17489725.2013.805828>
- Sak, H., Senior, A. & Beaufays, F. (2014), Long short-term memory recurrent neural network architectures for large scale acoustic modeling, *in* ‘Fifteenth annual conference of the international speech communication association’.
- Seidl, D. E., Jankowski, P. & Tsou, M.-H. (2016), ‘Privacy and spatial pattern preservation in masked gps trajectory data’, *International Journal of Geographical Information Science* **30**(4), 785–800.
- URL:** <https://doi.org/10.1080/13658816.2015.1101767>
- Snell, J., Swersky, K. & Zemel, R. (2017), Prototypical networks for few-shot learning, *in* ‘Advances in Neural Information Processing Systems’, pp. 4077–4087.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), ‘Dropout: a simple way to prevent neural networks from overfitting’, *The Journal of Machine Learning Research* **15**(1), 1929–1958.
- Ustinova, E. & Lempitsky, V. (2016), Learning deep embeddings with histogram loss, *in* ‘Advances in Neural Information Processing Systems’, pp. 4170–4178.
- Wagner, R. A. & Fischer, M. J. (1974), ‘The string-to-string correction problem’, *Journal of the ACM (JACM)* **21**(1), 168–173.
- Xiao, Z., Wang, Y., Fu, K. & Wu, F. (2017), ‘Identifying different transportation modes from trajectory data using tree-based ensemble classifiers’, *ISPRS International Journal of Geo-Information* **6**(2), 57.
- Yang, D., Zhang, D. & Qu, B. (2016), ‘Participatory cultural mapping based on collective behavior data in location-based social networks’, *ACM Transactions on Intelligent Systems and Technology (TIST)* **7**(3), 30.
- Yang, D., Zhang, D., Zheng, V. W. & Yu, Z. (2015), ‘Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns’, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **45**(1), 129–142.
- Zhao, W. X., Zhou, N., Sun, A., Wen, J.-R., Han, J. & Chang, E. Y. (2017), ‘A time-aware trajectory embedding model for next-location recommendation’, *Knowledge and Information Systems* pp. 1–21.
- Zheng, Y., Chen, Y., Li, Q., Xie, X. & Ma, W.-Y. (2010), ‘Understanding transportation modes based on gps data for web applications’, *ACM Transactions on the Web (TWEB)* **4**(1), 1.
- Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T. & Zhang, F. (2018), Trajectory-user linking via variational autoencoder., *in* ‘Proceedings of the 27th International Joint Conference on Artificial Intelligence’, pp. 3212–3218.
- Zhou, N., Zhao, W. X., Zhang, X., Wen, J.-R. & Wang, S. (2016), ‘A general multi-context embedding model for mining human trajectory data’, *IEEE transactions on knowledge and data engineering* **28**(8), 1945–1958.