A Shape-Based Map Matching Approach for Geographic Transferability of Discriminative Subtrajectories

Cristiano Landi[1,2,*], Riccardo Guidotti[1,2]

[1]University of Pisa, Pisa, Italy.
[2]ISTI-CNR, Pisa, Italy.

Abstract

This paper addresses the challenge of map matching and geographic transferability in trajectory analysis. Existing methods often face limitations tied to specific coordinates or road networks. In response, we propose GASM, a shape-based map matching method that relies solely on trajectory shapes, irrespective of geographic origin. GASM introduces a symbolic road network representation, facilitating efficient searches based solely on trajectory shapes. Our experimentation, spanning over 5,000 km of roads, demonstrates GASM's ability to accurately position trajectories with an impressive accuracy exceeding 90%. Notably, GASM stands as the first in the literature to perform shape-based symbolic map matching without prior knowledge of the geographic region.

Keywords

Map Matching, Geographic Transferability, Machine Learning, Discriminative Subtrajectories

1. Introduction

In recent years, the widespread adoption of cutting-edge technologies equipped with Global Positioning System (GPS) devices has enabled the recording of positions for various moving objects, ranging from cars and transportation vehicles to phones and wearables. Unfortunately, the coordinates captured by these sensors often fail to accurately reflect real positions due to physical constraints and/or legal regulations. Nevertheless, in various applications, it is imperative to accurately align GPS trajectories with a road network. For instance, in navigation services, map matching empowers drivers to monitor their exact locations and receive optimal routes to specified destinations. Conversely, in machine learning tasks, map matching enhances users' mobility information by incorporating knowledge related to the territory, such as Points Of Interest (POI), feature engineering [1, 2, 3, 4], or the identification of discriminatory subsequences, such as mobility shapelets [5, 6, 7]. Without an appropriate map-matching procedure, reliance on an expert becomes necessary to determine which features can be extracted from trajectories concerning the territory. However, the reliance on ad-hoc features restricts the applicability of machine learning methods and amplifies sensitivity to input changes [1], rendering it unsuitable for geographic transferability. This implies the challenge of extracting mobility patterns from one geographical region and effectively applying them in another region [8, 9].

Particularly noteworthy are recent advancements in machine learning leveraging shapelet-based subtrajectories [5, 6, 7]. Originating from the domain of time series analytics, shapelets represent discriminatory subsequences that encapsulate a collection of distinctive *shapes*, crucial for discerning specific classes [10]. Various approaches exist for defining discriminative subtrajectories. In [6], the Movelet method is introduced—an approach for extracting discriminative subtrajectories selected through a rigorous statistical test. During the discovery phase, Movelet generates candidate subtrajectories by extracting all possible subsequences with more than two contiguous observations, utilizing a sliding window. Building upon the foundation laid by Movelet, Geolet is introduced in [7]. This extension incorporates a normalization step after the discovery phase. The normalization step is designed to ease the comparison of discriminative subsequences with trajectories recorded in diverse geographical regions. The underlying rationale is that a subtrajectory pinpointing a sudden break in a road segment in one city should exhibit similarities to a subtrajectory associated with the same event in another city. This normalization enhances the method's adaptability across various geographic contexts.

While Geolet successfully addresses the limitation of Movelet by providing normalized subtrajectories, thereby enhancing geographic transferability independent of specific GPS coordinates, it introduces a potential vulnerability tied to the road network.

Our underlying hypothesis is that the less frequently a trajectory occurs, the greater the likelihood that shape-based methods utilizing it as a discriminative subse-

Published in the Proceedings of the Workshops of the EDBT/ICDT 2024 Joint Conference (March 25-28, 2024), Paestum, Italy
*Corresponding author.
cristiano.landi@phd.unipi.it (C. Landi);
riccardo.guidotti@unipi.it (R. Guidotti)
0000-0003-4907-9728 (C. Landi); 0000-0002-2827-7613 (R. Guidotti)

quence may not capture the intrinsic features of the trajectory but rather only its geographic position. In essence, if a discriminative subtrajectory is intrinsically linked to a particular road network due to its distinctive shape, it becomes unsuitable for geographic transferability. This limitation arises from the fact that the discriminatory aspect is not rooted in the movements themselves but rather in the structural characteristics of the road network. Consequently, evaluating the geographic transferability of discriminative subtrajectories necessitates a shape-based map-matching approach that exclusively relies on shapes without prior knowledge of the position. Regrettably, to the best of our knowledge, such an approach is currently unavailable. This underscores the need for innovative solutions in the realm of shape-based map matching to comprehensively assess the adaptability of discriminative subtrajectories across diverse geographical contexts.

To overcome this limitation, our paper introduces GASM, an Geographic Automaton Shape-based map Matching approach. GASM relies solely on the shape of a trajectory to accurately determine its position within the road network. Specifically, GASM employs a symbolic representation to transform the road network, constructing a spatial index independent of coordinates. This unique approach allows for efficient trajectory searches based solely on their shapes. To the best of our knowledge, GASM is the first proposal in the literature that exclusively utilizes a discretized representation of a trajectory's shape, devoid of any knowledge of the geographic region, for shape-based map matching. Our experimentation with GASM on a novel comprehensive geographical dataset spanning over 5,000 km of roads in Tuscany, central Italy, demonstrates its capability to identify correct alignments with an impressive accuracy exceeding 90%. Furthermore, GASM exhibits efficiency, as it can construct the necessary representation for the entire dataset in less than 1.5 hours, maintaining a linear complexity at query time.

The paper is organized as follows. Section 2 summarises the related works concerning map-matching methods and the challenges posed by geographic transferability. In Section 3, we encapsulate the technical concepts essential for comprehending the algorithm delineated in Section 4. The outcomes of experiments conducted with GASM are detailed in Section 5. Finally, Section 6 encapsulates our findings and delves into potential avenues for future developments.

2. Related Works

In the following, we provide a concise overview of the literature concerning map matching methods and elucidate the geographic transferability problem, introducing key strategies employed to tackle this challenge.

In the literature of trajectory analysis, a multitude of strategies exists for mapping trajectories onto a road network. For high-frequency sampled trajectories, the simplest approach involves associating each spatio-temporal point with the nearest street segment [11, 12]. However, these techniques, while fast and straightforward, have exhibited inaccuracies, particularly at intersections and parallel roads. To address these limitations, enhancements have been introduced, incorporating heading direction or employing a Kalman filter to eliminate outlier points in trajectories [13]. Alternatively, some approaches leverage probabilistic-based map-matching algorithms, integrating hidden Markov models to identify the most likely sequence of road segments aligning with the trajectory [14, 15]. On the other hand, in the context of low-sampled trajectories, much of the existing literature presupposes that the most probable route connecting two successive points is also the shortest or fastest [16]. However, in [17], is introduced a map-matching algorithm that exploits temporal intervals between GPS points. This method identifies the optimal match between two GPS points by selecting the route with the most similar travel time. Also, in [18] is proposed a method for map matching low-sampled trajectories based on supplementary information such as speed and moving direction, typically collected alongside spatial locations.

Geographic transferability encapsulates the challenge of extending knowledge gleaned from one geographic region to another. This entails constructing machine learning models capable of adeptly executing tasks in regions distinct from their original training grounds. The core of this challenge emerges from pronounced disparities in data distributions, patterns, and characteristics across diverse geographical locations. As articulated in [8, 9], models trained on data from one region may encounter difficulties in generalization when confronted with the unique variations inherent in another region. In pursuit of a global model, a fundamental strategy is employed as demonstrated in [8], where diverse data sources are aggregated on a global scale to build a transferable model. Conversely, in [9], city indicators are identified pertaining to road networks, traffic flows, and individual mobility, to facilitate the assessment of similarities between geographical regions. Subsequently, an ensemble classifier is devised, computing the output as a weighted average of outputs generated by individual local classifiers. Notably, the importance of local models is determined by their higher similarity to the target regions compared to others in the ensemble with respect to the city indicators. Alternate methodologies involve adapting a model initially trained in a data-rich region and transplanting it to a target region characterized by limited data availability. This adaptation process entails integrating additional data to compute sub-region similarities, subsequently enabling

the remapping of the model [19, 20].

3. Problem Setting

In this section, we articulate the fundamental concepts essential for comprehending our proposal. Initially, we establish a shared language and framework by introducing notation that serves as a basis for discussing key elements. Subsequently, we delve into the transformation facilitated by Geolet, a catalyst for the motivation behind this work. Ultimately, we present a formal introduction to the problem at hand.

Definition 1 (Trajectory)**.** A *trajectory* X is a sequence of spatio-temporal points $X = \{\vec{\mathbf{x}}_{t_0}, \dots, \vec{\mathbf{x}}_{t_m}\} \in \mathbb{R}^{m\times 3}$ where the spatial vectors $\vec{\mathbf{x}}_{t_j} = (\text{lat}_{t_j}, \text{long}_{t_j})$ are sorted by increasing time t_j, i.e., $\forall 1 \leq j < m$ we have $t_j < t_{j+1}$.

In a sense, trajectories can be viewed as multivariate time series containing two signals, i.e., the latitude and longitude, recorded at non-constant sampling rates [5, 21, 6]. In order to simplify notation, we will use j instead of t_j every time. A trajectory classification dataset is a set of trajectories with a vector of labels attached. Formally:

Definition 2 (Trajectory Dataset)**.** A *trajectory dataset* $\mathcal{X} \in \mathbb{R}^{n\times m\times 3}$ is a set of n trajectories, $\mathcal{X} = \{X_0 \dots, X_n\}$.

For simplicity, we use a single symbol m to denote the lengths of the trajectories, even if a dataset can contain trajectories with a different number of observations. Similarly, we emphasize that there is no constraint on the sampling rate, i.e., we can have a non-constant sample in the same trajectory. Furthermore, we define a *subtrajectory* as:

Definition 3 (Subtrajectory)**.** Given a trajectory X of length m, a subtrajectory $S = \{\vec{s}_j, \dots, \vec{s}_{j+l}\} \subset X$, of length $l \leq m$, is an ordered sequence of consecutive values such that $0 \leq j \leq m - l$.

As previously mentioned, Movelet [6] and Geolet [7] are shapelet-inspired [10] trajectory approaches that identify discriminative subtrajectories for classification purposes. They both select the most discriminative subtrajectories w.r.t. the target label using the *mutual information* [22]. Like shapelet-based approaches, Movelet and Geolet extract discriminative subtrajectories that can be used to train any machine learning model [23]. Indeed, once the most discriminative subtrajectories are identified, a trajectory dataset can be transformed into a tabular representation capturing the distance between trajectories and discriminative subtrajectories through the subtrajectory transform function:

Definition 4 (Subtrajectory Transform)**.** Given a dataset $\mathcal{X}$ and a set $\mathcal{S}$ containing h subtrajectories, the *subtrajectory transform* converts $\mathcal{X} \in \mathbb{R}^{n\times m\times 3}$ into a real-valued matrix $T \in \mathbb{R}^{n\times h}$, obtained by taking the *best fitting* of each trajectory $X \in \mathcal{X}$, and each subtrajectory $S \in \mathcal{S}$.

Movelet computes the *best fitting* as the minimal Euclidean Distance (ED) between S in each subsequence of length $l = |S|$ of X, formally,

$$bestfit_{Movelets}(X, S) = \min_{j=0}^{m-l}\{ED(X_{j:j+l}, S)\} \quad (1)$$

On the other hand, Geolet computes the *best fitting* in the same way, but geographically shifting S to overlap each subsequence of X of length l. In particular, Geolet extends the best fitting function of Movelet by adding a pre-processing function, *shift*, that subtracts the value of the first vector of the subsequence from all the others,

$$bestfit_{Geolet}(X, S) = \min_{j=0}^{m-l}\{ED\big(shift(X_{j:j+l}), shift(S)\big)\} \quad (2)$$

where $shift(X) = \{\vec{x}_0 - \vec{x}_0, \dots, \vec{x}_m - \vec{x}_0\}$. The *shift* function makes Geolet suitable for geographic transferability not being tide to the territory.

Finally, in order to present map matching we define a road network as follows:

Definition 5 (Road Network)**.** A road network $G = \langle V, E\rangle$ is a directed graph where $V = \{v_1, \dots, v_p\}$ is the set of p road junction (or nodes), and $E = \{e_1, \dots, e_q\}$ is the set of q road segments (or edges), where $e_i = (v_{i_1}, v_{i_2})$.

We underline that we rely on an enhanced road network representation where, for each edge, we also have access to the road segment geometries expressed as a sequence of k latitude and longitude, formally $shape(E) = \{\vec{x}_0, \dots, \vec{x}_k\}$ for some $E \in \mathcal{E}$. As for trajectories, for simplicity of notation, we use a single symbol k to denote the lengths of the points describing the geometry of the road segment, even if, in real-case scenarios, the shape can be described using an arbitrary number of points.

We are now able to formalize the *shape-based map matching problem* as follows:

Definition 6 (Shape-based Map Matching)**.** Given a road network $\mathcal{G} = \langle V, E\rangle$ and a trajectory X, the *shape-based map matching problem* consists in finding the best sequence of edges $Y = \{e_1, \dots, e_z\} \subseteq E$ such that does not exist another sequence of edges $Y' \subset E$ different from Y, i.e., $Y' \neq Y$, where $bestfit_{Geolet}(X, shape(Y')) < bestfit_{Geolet}(X, shape(Y))$.

In other words, the shape-based map matching problem involves determining the optimal alignment for a (sub)trajectory X within a designated road network G, relying on the configuration of the edges comprising the road segments. It is essential to emphasize that this map matching endeavor necessitates resolution without any reliance on GPS coordinates as the usage of the *shift* operator normalizes the trajectory X rendering state-of-the-art map matching methods unsuitable for this particular task.

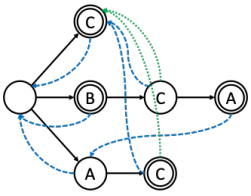

Figure 1: Aho-Corasick automaton using symbolic sequences $\{AC, B, BCA, C\}$. Blue dashed arches are suffix arches, while green dotted arches are the dictionary suffix arches.

4. Shape-based Map Matching

To tackle the shape-based map-matching problem, our aim is to design a map-matching method with the capability to accurately deduce the original GPS coordinates of a trajectory within a designated road network. Crucially, this precision is sought exclusively through an examination of the trajectory's shape and the configurations of the edges within the road network, entirely independent of any reliance on GPS coordinates.

A brute-force approach to address the problem involves map matching all conceivable alignments of X within every segment E of the road network G, employing the $bestfit_{Geolet}$ function. However, this naive strategy is only viable for small road networks due to the algorithmic complexity being $O(|E|(m-k)k)$, where m and k represent the number of points characterizing the trajectory X and the number of points describing each road segments in E, respectively[1]. This limitation also extends to other map-matching algorithms that rely on latitude and longitude coordinates to confine the matching scope to the nearest roads.

We overcome this limitation by proposing GASM a Geographic Automaton Shape-based map Matching approach that is able to significantly reduce the number of road segment alignments to test with the brute force method. In essence, GASM comprises two key steps. Initially, leveraging the Aho-Corasick algorithm [24], GASM constructs a shape-based index for all road segments in E, portraying it as a geographic finite state automaton. Subsequently, GASM facilitates querying the automaton to pinpoint a set of candidate partial matches between X and $\{shape(Y) | \forall\ Y \subseteq E\}$. Further elucidation of these two steps is provided in the subsequent sections.

Geographic Automaton Construction. GASM leverages the Aho-Corasick algorithm to construct a geographic automaton, serving as a spatial index for expedited query processing [24]. The Aho-Corasick algorithm, renowned for string searching, takes a set $\mathscr{W} = \{W_1, \ldots, W_n\}$ as input, where each W_i represents a finite sequence of symbols over an alphabet Σ. Subsequently, it builds a finite-state automaton based on the sequences in $\mathscr{W}$ within a given finite symbol sequence defined over the alphabet Σ. Consequently, the automaton, constructed using the dictionary $\mathscr{W}$, identifies a subset $\mathscr{W}' \subset \mathscr{W}$ wherein each sequence in $\mathscr{W}'$ is contained in Q. Figure 1 provides an illustration of the automaton created by the Aho-Corasick algorithm, using the sequences $\mathscr{W} = \{AC, B, BCA, C\}$ over the alphabet $\Sigma = \{A, B, C, D\}$. The Aho-Corasick algorithm initiates by constructing a suffix trie [25], depicted in black in Figure 1. Subsequently, it designates all leaves of the trie as final states of the automaton and introduces edges to complete the automaton. Two types of edges are incorporated, connecting their respective suffixes: *suffix edges*, depicted in blue, are utilized in the case of a mismatch, without guaranteeing that the suffix is also a sequence in the dictionary. In contrast, *dictionary-suffix edges*, portrayed in green, guarantee that the suffix is a sequence present in the dictionary. These operations unfold linearly concerning the total number of symbols in the input dictionary $\mathscr{W}$. The automaton enables the search for all sequences contained in a query by traversing the automaton, achievable in linear time relative to the query's length.

[1]For each $Y \subseteq E$, we compute the Euclidean Distance (linear complexity), for all the possible $m-k$ alignments of $shape(Y)$ in X.

Algorithm 1: $build(E, d, \Sigma, h)$

Input : E - road segments, d - resampling distance, α - max nbr. of symbols, h - h-hop aggregation
Output : *aho* - Aho-Corasick automaton

$E_h \leftarrow aggregate(E, h)$; // aggregate trajectories
$\mathscr{W} \leftarrow \emptyset$;
for $e \in E_h$ **do** // for each road segment
  $X \leftarrow resample(shape(e), d)$; // resample traj.
  $\vec{X} \leftarrow direction(X)$; // get traj. direction
  $W \leftarrow SAX(\vec{X}, \alpha)$; // discretize traj.
  $\mathscr{W} \leftarrow \mathscr{W} \cup \{W\}$; // add to dict.
return $Aho\text{-}Corasick(\mathscr{W})$;

Algorithm 2: $search(X, A, d, \Sigma)$

Input : X - query traj., A - Aho-Corasick automaton, E - road segments, d - resampling dist., α - max nbr. of symbols
Output : Y^*, $\mathscr{Y}$ - best methc and best candidates

$X' \leftarrow resample(X, d)$; // resample traj.
$\vec{X}' \leftarrow direction(Q)$; // get traj. direction
$Q \leftarrow SAX(\vec{X}', \alpha)$; // discretize using SAX
$\mathscr{Y} \leftarrow search(A, Q, E)$; // get best candidates
$Y^* \leftarrow \arg\min_{Y \in \mathscr{Y}} bestfit_{Geolet}(Y, X')$;
return Y; // return the best match

Algorithm 1 delineates the procedural steps requisite of GASM for constructing the Aho-Corasick automaton. The algorithm accepts, as input, the road segments E of the road network G, the resampling distance d, the maximum allowed number of symbols α, and the number of hops h, producing a geographical automaton A as output. The GASM-*build* algorithm begins by aggregating the road network, concatenating h times a road segment to linked road segments in E_h to extend the length of existing segments and enhance their representativeness (line 1). Subsequently, it initializes an empty dictionary $\mathscr{W}$ (line 2). The following steps are applied for each road segment in E_h (denoted as e). Given that the shape of a road segment e may be described by varying numbers of points based on its length and sinuosity, GASM initially resamples the geometries into a series of evenly spaced points X. This ensures that the symbolic representation's length, crucial for Aho-Corasick automaton construction, is proportional solely to the road length. To fulfill the prerequisite of representing each road segment e in a discretized space, a sequence of symbols is generated (line 5). Subsequently, GASM determines the heading direction $\vec{X}$ between consecutive points along the resampled road segment, transforming the shape of each road sequence e into a univariate time series of directions $\vec{X}$ with a consistent length-based sampling rate d (line 6). This facilitates the utilization of Symbolic Aggregate approXimation (SAX) [26] to obtain a symbolic representation of each road segment over an alphabet Σ (line 7). These representations are added to the dictionary $\mathscr{W}$. Finally, the dictionary of discretized representations of the road segments is employed to construct the Aho-Corasick automaton, which is then returned as the output (line 8).

Shape-based Matching. Once the construction of the geographic automaton is complete, GASM can execute shape-based map matching over the automaton following the steps outlined in Algorithm 2. GASM-*search* takes as input the query trajectory X, the geographical automaton A, the road segments E, the resampling distance d, and the maximum allowed number of symbols α. It yields the sequence of edges $Y \subseteq E$ that minimizes the $bestfit_{Geolet}$ function, as per the ensuing procedure. The initial three steps of Algorithm 2, aligning with Algorithm 1, involve resampling the query trajectory X, extracting its direction, and transforming it into a symbolic representation Q. Indeed, the same preprocessing applied to the road segments E is applied to the query trajectories. Subsequently, the geographic automaton A is utilized to perform a linear search for the best matches $\mathscr{Y}$ among all possibilities offered by E (line 4). This implementation enables GASM to identify an "initial best match", presenting a set of best match candidates $\mathscr{Y} = \{Y_1, \ldots, Y_n\}$. From this set, the final selection of the optimal alignment Y^* is determined through a naive approach (line 5).

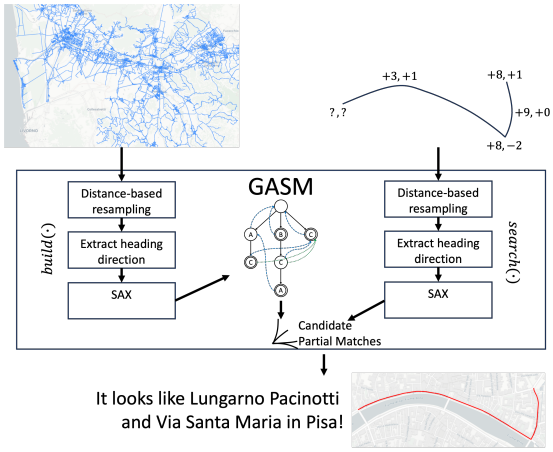

Figure 2: Summary of GASM, depicting geographic automaton construction (left) and shape-based matching (right).

	Hyperparameter GASM	Values
d	Resampling distance (m)	[5, 10, 20, 50]
$\lvert\Sigma\rvert$	Alphabet size	[4, 8, 16]
h	h-hop aggregation	[0, 1, 2, 3]

Table 1
Tested hyperparameters with their values.

Figure 2 visually summarise GASM. On the left side, the geographic automaton construction phase is depicted, wherein each road within an arbitrary large road network is indexed according to its heading direction. On the right side, the shape-based matching phase is illustrated. Here, given a trajectory with known shape but unknown origin point, GASM computes the set of potential partial map matches. Subsequently, it selects the match that minimizes the $bestfit_{Geolet}$ function.

5. Experiments

In this section, we evaluate the effectiveness of GASM[2]. First, we present the experimental setting, then we report and discuss the best performance achieved. Finally, we illustrate details of the hyperparameter tuning and the result of a sensitivity analysis w.r.t. some data properties.

Experimental Setting. Regrettably, only a handful of mobility datasets, such as GeoLife and Porto Taxi[3], are available as open access [1, 7]. However, these datasets possess limited geographic coverage, rendering them unsuitable for our study. Thus, we introduce a novel high-sample rate dataset derived from the publicly acces-

[2]Python code: https://t.ly/wVlXS. We ran our experiments on a *2x*Intel Xeon Gold 6342 24-core CPU, limiting each test to use at most 12 cores.
[3]GeoLife: https://t.ly/6VJ-E. Porto: https://t.ly/0GMR9.

Province	#Trj	Length (km) Totoal	Length (km) Average (σ)	Length (#points) Average (σ)	Kind of Road (%) Motorway	Trunk	Primary	Secondary	Minor
Arezzo	17	341	17.2 (18.38)	853 (851)	0.019	0.227	0.205	0.344	0.196
Firenze	58	1041	30.4 (37.31)	1526 (1427)	0.122	0.032	0.392	0.205	0.249
Grosseto	35	215	6.7 (9.35)	578 (545)	0.020	0.133	0.054	0.327	0.466
Livorno	53	410	18.5 (25.42)	916 (645)	0.410	0.043	0.089	0.144	0.313
Lucca	39	804	17.5 (15.49)	1128 (1133)	0.225	0.000	0.056	0.442	0.278
M. Carrara	46	267	5.1 (3.54)	625 (430)	0.187	0.173	0.160	0.267	0.212
Pisa	35	831	26.0 (23.04)	1347 (1178)	0.000	0.002	0.219	0.200	0.578
Pistoia	20	468	53.1 (31.07)	929 (777)	0.000	0.186	0.251	0.163	0.399
Prato	31	146	4.5 (2.37)	660 (672)	0.141	0.050	0.395	0.146	0.269
Siena	24	497	22.4 (16.54)	983 (660)	0.557	0.032	0.340	0.026	0.046

Table 2
Dataset description. Besides average values are reported standard deviations (σ).

Province	Method Performances Selectivity Factor ↓	Accuracy ↑	Building Time (s) ↓	Road Network Characteristics #Road Segments	#Intersections	Avg node Degree	Length (km)
Arezzo	0.096	0.875	708.32	133028	54485	4.883	26852
Siena	0.068	1.000	516.37	175088	72079	4.858	28949
Pistoia	0.080	0.950	433.24	81870	34492	4.747	12363
Lucca	0.070	0.846	658.03	141149	59274	4.763	20328
Firenze	0.399	0.263	157.17	299312	119068	5.028	39025
Grosseto	0.060	0.823	421.66	121045	50014	4.841	26818
Livorno	0.069	1.000	277.38	96631	39613	4.879	11269
M. Carrara	0.056	0.978	294.50	71917	300071	4.783	10809
Pisa	0,081	0.857	1007.06	150954	62580	4.824	22396
Prato	0.105	0.936	190.77	45060	18794	4.795	5224
Macro Avg (σ)	0.108 (0.103)	0.853 (0.217)					

Table 3
Selectivity factor, accuracy, automaton construction runtime, and other road network informations.

sible 2013 GPS traces on OpenStreetMap[4]. Although the initial OpenStreetMap dataset encompasses GPS trajectories spanning the entire globe, our analysis concentrates on the ten provinces in Tuscany, a region encompassing $22,985 km^2$ in central Italy. The Mappymatch python package[5] was employed to map-match each trajectory, retaining only those trajectories with an average error of less than $10m$. The final dataset encompasses 358 distinct trajectories, covering a total travel distance of $5,024km$ and described by 300,049 GPS points. Additional information on the types of roads traversed in each province in Tuscany, as per the OpenStreetMap taxonomy[6], is presented in Table 6.

Within the framework of our shape-based map-matching formulation, we aim to address the following questions. First, to what extent can GASM infer the original GPS coordinates without utilizing them for map matching? Second, how effectively can GASM reduce the number of potential alignments compared to the entire road network? The first question is evaluated through the metric of *accuracy*.On the other hand, the evaluation of the second question relies on the metric of *selectivity*, commonly employed in database literature [27]. Selectivity measures the reduction of potential alignments between a query result and the entire dataset. In our context, selectivity is defined as the ratio of matched road segments ($|\mathcal{Y}|$) to the total number of road segments ($|E|$). For accuracy, higher values indicate better results, while for selectivity, lower values indicate better outcomes.

GASM Performance. Table 5 presents the performance metrics of GASM across individual provinces. To determine the optimal hyperparameters, a grid search was conducted over the values outlined in Table 6, specifically for the province of Grosseto. This process yielded the following hyperparameters: a resampling distance of $d = 10$ meters, an alphabet size of $\alpha = 8$ symbols, and a street aggregation of $h = 2$ hops. GASM showcases an impressive ability to deduce the original GPS coordinates, achieving an average light accuracy of 90.1%. Furthermore, it significantly narrows down the potential alignments, as indicated by the selectivity factor, reducing it to just 10.8% of the original road network. These commendable results are attained while maintaining a

[4]OpenStreetMap 2013 public GPS traces: https://t.ly/q7u2N
[5]Mappymatch: https://t.ly/RHafS
[6]Highway taxonomy: https://t.ly/NpxZv

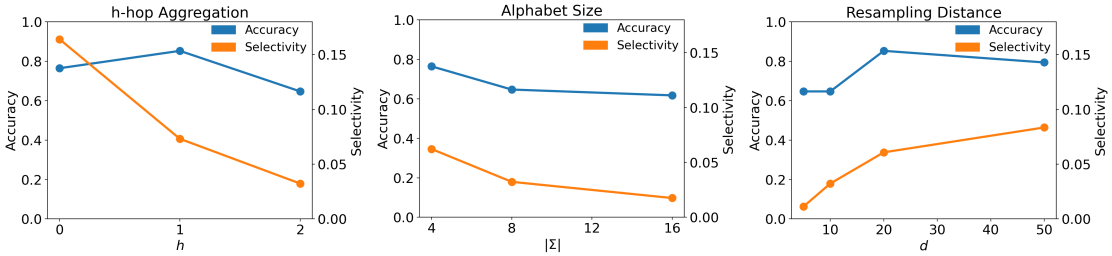

Figure 3: Hyperparameters influence: h-hop aggregation (left), α alphabet size (center), and d resampling distance (right).

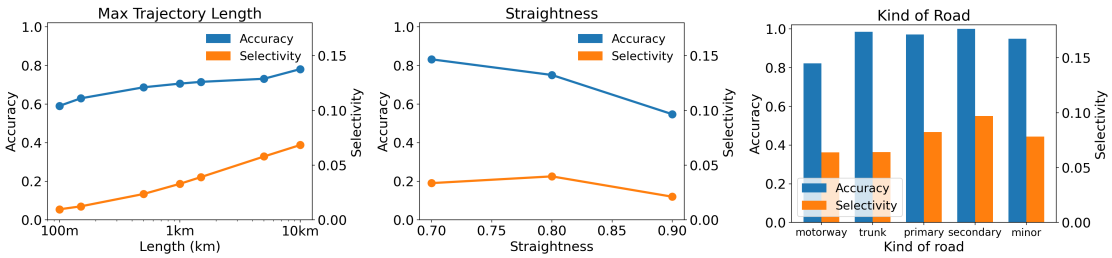

Figure 4: Influence of trajectory length (left), trajectory straightness (center), and kind of road (right) on performance metrics.

reasonable automaton construction time of 7.8 minutes per province, resulting in a total indexing time of a mere 1.29 hours for the entire region.

Hyperparameters Tuning. In this section, we present the results of experiments conducted on the province of Grosseto while varying the hyperparameters detailed in Table 6. Initially, we compute the Pearson correlation between the method's hyperparameters and two key performance metrics: the selectivity factor and accuracy. Figure 3 visually depicts the changes in performance metrics, emphasizing variations in the top three most influential hyperparameters—those exhibiting the highest absolute values of Pearson correlation. Notably, the most influential hyperparameter is the number of road segment aggregations (h), demonstrating a correlation of -0.52 with the selectivity. Thus, increasing h proves beneficial for GASM as it helps select fewer candidate road segments without significantly impacting accuracy. The alphabet size (α) displays correlations of -0.44 and 0.40 with respect to the selectivity and accuracy, respectively. This hyperparameter introduces a trade-off, as increasing the number of symbols reduces selectivity but may lead to a slight decrease in accuracy. Finally, the resampling distance (d) exhibits a correlation of 0.22 with accuracy. Interestingly, decreasing d slightly enhances accuracy according to our observations.

Sensitivity Analysis. We delve here into the variations in performance with respect to the length of the query trajectory X. Additionally, we explore the discriminative nature of trajectories based on the type of road. Our hypothesis posits that straight streets, such as motorways, exhibit lower discriminative characteristics. Consequently, trajectories observed on such roads are more likely to avoid re-identification, suggesting enhanced geographic transferability. To investigate this, we identify the type of road traveled within each segment. In cases where multiple types of roads are encountered, we perform a majority vote weighted by road length. Additionally, to examine the influence of changes in the input data, we create random subtrajectories of varying lengths, including 100m, 150m, 500m, 1km, 1.5km, 5km, and 10km, derived from our OpenStreetMap dataset. In order to assess our hypothesis, we evaluate the *straightness* [4] of each subtrajectory by calculating the ratio between the shortest path from the origin to the destination and the actual trajectory.

Figure 4 encapsulates these results. The initial plot on the left highlights a notable trend: an increase in subtrajectory length correlates with a rapid elevation in both accuracy and selectivity. In simpler terms, as the subtrajectory length extends, the model's precision improves. The central plot reveals that trajectory straightness has a negligible impact on the number of candidate matches. However, as trajectories become more linear, the accuracy experiences a decline. Finally, the rightmost plot illustrates the method's performance across various road types. This plot validates the findings of the straightness plot: roads with greater straightness, like motorways, pose the greatest challenge for re-identification. Con-

versely, more sinuous roads present a slightly higher difficulty in re-identification, reflected in a higher selectivity but with a concomitant boost in accuracy.

6. Conclusion

In this paper we have introduced GASM, a map matching method capable of determining a trajectory's position solely based on its shape. Our experiments showcase that GASM significantly reduces the number of potential alignments and deduces the original GPS coordinates with remarkable accuracy. Further analysis reveals that longer and less linear trajectories are more straightforward to map match. However, this observation raises concerns about the potential for shape-based methods to inadvertently learn geographic positions instead of focusing on other intrinsic features. As a part of future work, as outlined at the beginning, we aim to assess the geographic transferability of shape-based methods, such as Geolet, by incorporating GASM. Specifically, we propose giving more weight to the selection of discriminative subsequences with higher selectivity rather than basing the decision solely on a statistical test.

Acknowledgments

This work is partially supported by the EU NextGenerationEU programme under the funding schemes PNRR-"SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, and *GreenDatAI* G.A. 101070416.

References

[1] C. L. da Silva, et al., A survey and comparison of trajectory classification methods, in: BRACIS, IEEE, 2019, pp. 788–793.

[2] A. Bolbol, et al., Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification, CEUS 36 (2012) 526–537.

[3] S. Dodge, et al., Revealing the physics of movement: Comparing the similarity of movement characteristics, CEUS 33 (2009) 419–434.

[4] P. J. Almeida, et al., Indices of movement behaviour: conceptual background, effects of scale and location errors, Zoologia (Curitiba) 27 (2010) 674–680.

[5] I. Kontopoulos, et al., Traclets: Harnessing the power of computer vision for trajectory classification, arXiv:2205.13880 (2022).

[6] C. A. Ferrero, et al., MOVELETS: exploring relevant subtrajectories for robust trajectory classification, in: SAC, ACM, 2018, pp. 849–856.

[7] C. Landi, et al., Geolet: An interpretable model for trajectory classification, in: IDA, volume 13876 of *LNCS*, Springer, 2023, pp. 236–248.

[8] F. Veronesi, et al., Assessing accuracy and geographical transferability of machine learning algorithms for wind speed modelling, in: AGILE, LNGC, 2017.

[9] M. Nanni, et al., City indicators for geographical transfer learning: an application to crash prediction, GeoInformatica 26 (2022) 581–612.

[10] J. Lines, et al., A shapelet transform for time series classification, in: ACM SIGKDD, 2012, pp. 289–297.

[11] C. White, et al., Some map matching algorithms for personal navigation assistants, TRC 8 (2000).

[12] D. Bernstein, et al., An introduction to map matching for personal navigation assistants (1996).

[13] M. A. Quddus, et al., A general map matching algorithm for transport telematics applications, GPS solutions 7 (2003) 157–167.

[14] Y. Liu, Z. Li, A novel algorithm of low sampling rate GPS trajectories on map-matching, EURASIP 2017 (2017) 30.

[15] X. Liu, et al., A st-crf map-matching method for low-frequency floating car data, TITS 18 (2016) 1241–1254.

[16] L. Jiang, et al., From driving trajectories to driving paths: a survey on map-matching algorithms, CCF TPCI 4 (2022) 252–267.

[17] P. Cintia, M. Nanni, An effective time-aware map matching process for low sampling gps data, arXiv preprint arXiv:1603.07376 (2016).

[18] G. Hu, et al., If-matching: Towards accurate map-matching with information fusion, TKDE 29 (2016) 114–127.

[19] L. Wang, et al., Smart city development with urban transfer learning, Computer 51 (2018) 32–41.

[20] L. Wang, et al., Cross-city transfer learning for deep spatio-temporal prediction, in: IJCAI, ijcai.org, 2019, pp. 1893–1899.

[21] R. Trasarti, et al., Myway: Location prediction via mobility profiling, Inf. Syst. 64 (2017) 350–367.

[22] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to data mining, Pearson Education India, 2016.

[23] A. J. Bagnall, et al., The great time series classification bake off, CoRR abs/1602.01711 (2016).

[24] A. V. Aho, et al., Efficient string matching: An aid to bibliographic search, ACM 18 (1975) 333–340.

[25] E. M. McCreight, A space-economical suffix tree construction algorithm, JACM 23 (1976) 262–272.

[26] J. Lin, et al., A symbolic representation of time series, with implications for streaming algorithms, in: DMKD, ACM, 2003, pp. 2–11.

[27] S. Acharya, et al., Selectivity estimation in spatial databases, in: SIGMOD, ACM, 1999, pp. 13–24.