UNIVERSITÀ
DEGLI STUDI
DI PALERMO

Detecting Similarities in Mobility Patterns

Article

Accepted version

P. Cottone, M. Ortolani, G. Pergola

In Proceedings of the Eighth European Starting AI Researcher Symposium (STAIRS 2016), The Hague, The Netherlands, August 29-30, 2016

It is advisable to refer to the publisher's version if you intend to cite from the work.

Publisher: IOS Press

Detecting Similarities in Mobility Patterns

Pietro COTTONE [a,1], Marco ORTOLANI [a] and Gabriele PERGOLA [a,2]
[a] *DICGIM – University of Palermo, Italy*

Abstract. The wide spread of low-cost personal devices equipped with GPS sensors has paved the way towards the creation of customized services based on user mobility habits and able to track and assist users in everyday activities, according to their current location.

In this paper we propose a new approach to extraction and comparison of mobility models, by means of the structure inferred from positioning data. More specifically, we suggest to use concepts and methods borrowed from Algorithmic Learning Theory (ALT) and we formulate mobility models extraction in term of Grammatical Inference (GI), an inductive process able to select the best grammar consistent with the samples and to provide multi-scale generative models. Moreover, we propose a similarity measure by adapting a state-of-the-art metric originally conceived for automata.

A thorough experimental assessment was conducted on the publicly available dataset provided by the *Geolife* project. Results show how a structural model and similarity metric can provide a better insight on data despite its complexity.

Keywords. structural knowledge, mobility models, grammatical inference

DRAFT

1. Introduction

During the past years, automated systems for the acquisition and processing of users' movements in everyday life has attracted growing attention, also thanks to the wide diffusion of cheap and commonly available devices (e.g. smartphones or GPS loggers) that can readily provide the location of their owners.

Human movements can be considered as schemata naturally induced, and heavily influenced by everyday routine [1]; indeed, the majority of location data regard the most frequent paths, such as those involving routes from home to work, or to recreational places; moreover, people are creatures of habit, who are prone to repeat often the same course of actions, hence the same itineraries. It is thus possible to reason in terms of *social patterns*, i.e. sequences that characterize human behaviors, as the main goal of mobility data analysis.

Identifying such patterns may however prove quite challenging, as it is hard to find meaningful features for a valid metric to evaluate and compare paths and mobility models. The massive collections of data nowadays readily available for machine processing

[1]Corresponding Author: Pietro Cottone, DICGIM – University of Palermo, Italy; E-mail: pietro.cottone@unipa.it

[2]This work was partially supported by the Italian Ministry of Education, University and Research on the "StartUp" call funding the "BIGGER DATA" project, ref. PAC02L1_0086 – CUP: B78F13000700008.

have further complicated the nature of the problem, and researchers are more and more aware that "measuring" does not seamlessly translate into "understanding". In the field of mobility and location data analysis, these considerations have appeared in full clarity due to the particular nature of the phenomenon: user movements are bound to adhere to schemas naturally induced from their daily routine, but it is not easy to make sense of data by letting models naturally *emerge* from the collected samples, as opposed to deducing them from pre-set assumptions.

Although the intrinsically structural nature of movements is intuitively evident, and may be empirically pointed out [2], current state-of-the-art systems focus only on the statistical analysis of path features, or on the description of their shapes by means of geometric tools [3]. A more promising approach would be instead to mine the *structural information* embedded into mobility data, which has only marginally be investigated in a few works in literature [4,5] that relied on formal languages and automata as their reference framework. As will be described in the following sections, our system explicitly uses symbolic encoding for user movements, subsequently processing them through a grammar inference algorithm that extracts the most relevant paths, and code them as a set of finite automa. The obtained model may thus be regarded as a representation of the *language of paths* for a user. Finally, we propose a new similarity metric for mobility models, not relying on any a-priori assumption or knowledge, but only on the features of the extracted model. The main contributions of this paper is fourfold and consists of (i) an effective strategy to encode real spatial data as strings of a formal language; (ii) a multi-scale, syntactic, mobility model; (iii) the customization of the inference process, through the adoption of an algorithm able to deal with noise and statistical relevance of paths; (iv) finally, a similarity metric exploiting properties of automata, explicitly aimed at detecting shared regularities in user habits.

The remainder of the paper is organized as follows. Section 2 summarizes the state-of-the-art systems for mobility models, while Sections 3 and 4 present our approach based on *Grammatical Inference* (GI). The description of a case-study application to mobility data follows in Section 4. Results of our experimental assessment will be shown in Section 5, and, finally, we will present our conclusions in Section 6.

DRAFT

2. Related work

A mobility model is a concise representation of user movements, synthesizing already observed paths in order to predict the future ones. The presence of regularities has motivated researchers to dive into the design of systems able to predict mobility behaviours, as demonstrated by the extensive literature produced in this research field [1,6]. The obtained results have shown that mobility models can be profitably applied to a wide range of scenarios, including user activity recognition, route prediction, extraction of *Points of Interest*, and location recommendation, anomaly detection or simulators [5]. *Location-Based Services* (LBSs), in particular, aim at exploiting information about location in order to deliver customized service to users, supporting them in carrying out everyday activities, or improving their travel experience [7]. Clearly, LBSs are heavily based on recommender systems, that need to effectively compare several users, pointing out their similarities and highlighting emergent behaviours, in order to predict relevant events or phenomena [8]. In this setting, the main problem is represented by the identification of a

successful metric to classify and compare users, according to the information embedded into raw data [9]. More recently, LBSs have appeared, and represent one of the most promising application scenario for mobility data. For example, in [10], authors show how mobility data is correlated with social networks; they propose a mobility model based on a combination of periodic and short-range movements, inferred by positioning data, and long-distance movements, based on social data analysis.

In all cases, a crucial aspect is represented by mobility model comparison, and the definition of a reliable similarity measure is one of the most important open issue in mobility data analysis. A framework to model user location history and mine similarities among different users is proposed in [11]. It is based on the analysis of the sequence of user movements, the popularity of each region visited and a hierarchical partitioning of the geographic area. The similarity measure is based on the longest common sequences between paths of two users, combined with the popularity of the region they are made of. An improvement of the previous work is proposed in [12], by the introduction of the semantic location history, that translate positioning data into semantic locations, i.e. shopping malls, cinema, restaurants, etc. This information is used to match paths of different users, in order to find a better correspondence with respect to their habits. The idea of using semantic information to empower similarity measure is proposed also in [13]; next location of a user path is predicted considering both the geographic and semantic features of other users trajectories. The main contribution of this work is a clustering strategy to measure similarity among users, based on the *Maximal Semantic Trajectory Pattern Similarity*, that compute the similarity between two trajectories counting their common parts, exploiting their longest common sequences. In [14], authors proposes a new approach in order to extract mobility models and measure similarity between users. In particular, they describe paths as temporal-annotated sequences of relevant locations, extending the idea proposed in [15]. Moreover, they improve the similarity measure proposed in [13], introducing a likelihood among the semantic for locations and taking into account temporal annotations to describe paths through T-patterns. In [16], similarity metric is based on routine activities, i.e. repeating activities at certain locations with regular time intervals. In a a first step, routines are extracted from daily trajectories by a clustering method; then, user similarities is calculated as weighted mean between their routine activities. The main limitation of the majority of the proposed solutions is that they do not take advantage of the natural recursive structure of paths, and rather opt to represent them by relying only on statistical properties or geometric description [3], often ending up in a overcomplicated and inflexible representation. They thus miss the fact that human travelling behavior can be described in different ways at varying spatiotemporal scales, as shown by [2].

On the other hand, some approaches have attempted to create structural models through a syntactic approach and a symbolic encoding, using formal grammars to represent target models. In [17], *Finite State Automata* were used to model mobility behaviors. Authors propose two approaches: in the first, the alphabet is made up of the "status" of the user and the states of the automaton are the locations (e.g., at home, at work); in the second one, the role of locations and "status" are switched. Locations were inferred through unsupervised learning algorithms, mining the most visited places; "status" categories are extrapolated from temporal sequences of movements. An approach based on grammar induction to analyze spatial trajectories was investigated in [4]. A grammar induction algorithm, called *mSEQUITUR*, was proposed; it is able to obtain a grammar

DRAFT

rule set from a trajectory for motif generation. Moreover, the authors present the *Trajectory Analysis and VIsualization System* (STAVIS), a trajectory analytical system that derives trajectory signatures and allows to extract relevant information from them, using a grammar inference algorithm. These approaches based on syntactic descriptions do not link the recursive nature of path with the description of movements. In particular, their representation are not able to deal with raw positioning data and require a very complex preprocessing in order to turn locations into symbols.

In the following, we will show how formal languages can be used in order to create multi-scale models, combining them with a recursive representation of coordinates.

3. Inference of User Mobility Patterns

Our approach aims at modelling user mobility habits from frequent paths expressed as sequences of locations. In particular, we assume that user mobility models can be described through formal languages; in other words, an unknown language describing mobility data is supposed to exist; we aim at uncovering this hidden structure and we choose to address this problem by GI. However, data collected in a real-life scenario is often in a numerical form, embedded in a geometric space, whose dimensions are the features selected by the designer [18]; thus it can not be promptly elaborated by GI algorithms, for which data in symbolic form is required instead. Thus, a preprocessing step is needed in order to perform symbolic encoding; as we will show in the next section, we rely on geohash encoding to turn locations into symbols. Once this step is accomplished, model extraction can be addressed as an instance of regular language inference.

Many algorithms have been proposed in literature, coping with learning from unbounded *symbolic* sequences remains an open challenge. In our formulation we will refer to *Algorithmic Learning Theory*, and, more specifically, we will make use of formal regular languages to represent user paths, and will focus on the recognizers of such languages, i.e. DFAs. The problem of model extraction thus corresponds to the inference of the most general recognizer (the *minimal Deterministic Finite Automaton* (DFA)) consistent with the given data, and can be addressed trough the process known as GI. Several approaches have been proposed in literature, all of which are based on the concept of *identification in the limit* initially formulated by Gold [19], who stated that the learning algorithm should identify the correct hypothesis on every possible data sequence consistent with the problem space.

The focus is on the way data is fed to the learning algorithm; for instance, the available data may consist of positive examples only, i.e. strings that are known to belong to the target language; this is the typical case in a setting of text analysis where, by definition, the strings used for training belong to the language: it is the so-called presentation *from text*. Such assumption is very stringent: as Gold proved, positive examples alone are not sufficient to ensure identification in the limit for any language of practical interest.

The alternative approach is called presentation *from informant*, whenever we are provided with negative examples too. Angluin proposed a theoretical framework known as Active Learning [20], which assumes the presence of an *oracle*, i.e. a “black box” knowing the entire target language and able to answer two kinds of queries, namely: membership, or equivalence queries. In the first case, the oracle will tell whether a string belongs to the language or not; on the other hand, an equivalence query is formulated

DRAFT

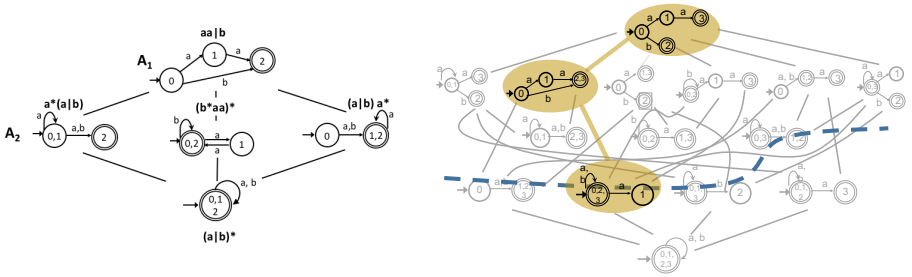


Figure 1. An example of a merging operation (left), with a sketch of a search in the induced state space (right).

by providing the oracle with a hypothesis (e.g. a DFA) to which it responds either by accepting it, thus indicating successful termination, or else by producing counterexamples (strings which the current DFA fails to accept) that may be used to improve the hypothesis. It may be proven that a complete presentation is sufficient to identify a regular language in the limit, but it remains hard to build an oracle in most practical scenarios, where the target language is not completely known in advance (otherwise, the model would probably be already known), and rather the given data consists of just a subset of positive and negative examples.

An alternative proposal was the formulation of the problem as a state space search. Starting with an automaton representing the available examples, the aim is to have it evolve toward a more general one, able to capture more strings of the unknown language. In order to fulfill this goal, a simple consideration may be of help: any automaton obtained by merging two states of a parent automaton accepts a language that is a superset of the original one (see the leftmost part of Figure 1 for a clarifying example). This provides a simple way to move from the recognizer of a specific language to a more general one, and is the basis for a class of inference methods known as *state merging* algorithms. The issue now regards how to prevent the more general language from including invalid strings; in other words, we need to limit overgeneralization, which would otherwise lead us to eventually merge all states into a single one, thus producing the automaton accepting all possible strings over the given alphabet. One possibility is to begin by building the *Prefix Tree Acceptor*, which accepts all available positive examples, and to proceed by producing all automata generated by merging all possible pairs of states, paying attention to reject all those that would mistakenly accept negative examples. It may be shown that all the automata that would become invalid as a consequence of further state merging are part of a "frontier" in the search space; the goal is to identify the most general automaton within those belonging to the frontier (see Figure 1, right)[3].

It is immediate to recognize the possible combinatorial explosion to which such inference algorithms are subject; moreover, since the search is limited only by negative samples, it is essential that they are as representative as possible of the target language; finally, the order in which they are presented to the inference algorithm is important.

One of the first documented algorithms in this context was *Regular Positive and Negative Inference* (RPNI) [23], which essentially attempted exhaustive search of the automata space toward the frontier, and was proven to be able to identify in the limit the minimum consistent automaton provided that the learning sample is *structurally com-*

[3] A thorough analysis of merge operator can be found in [21,22].

plete[4]. The addition of a heuristic used to limit the search to the parts of the automata space which arguably might contain the solution produced *Evidence-Driven State Merging* (EDSM) [24]. In a real application scenario, this approach suffers from some limitations; first of all, nearly always the target automaton is not available so it is not possible to check whether the learning sample does in fact contain a characteristic set; secondly, if data is affected by noise, the final DFA may be significantly compromised, because of the very nature of DFAs that do not allow to specify a degree of acceptance or rejection; in such cases, the algorithm typically yields a larger and less accurate DFAs. Such considerations strongly suggest the importance of trying to discard examples that might lead to overfitting, as well as to exploit their distribution to detect which are the relevant ones.

3.1. Dealing with Imprecise or Incomplete Data

The attempt to improve EDSM gave rise to *Blue** [25], which is based on a clever strategy to deal with a high amount of data and whose key insight is a statistical distinction between relevant and irrelevant information, which is treated as noise. Among the different types of noise that can be observed (e.g. noisy labels, incomplete data, fuzzy data, etc.), the case of mislabeled data is addressed; in other words, the algorithm assumes that some positive examples might be mistakenly regarded as negative, and vice versa.

Following the "features selection" terminology [26], at least two different approaches are possible: either irrelevant examples are removed *before* inference begins (which is known as *filter* strategy), or noisy samples must be detected and processed *during* the inference process (*wrapper* strategy). The first approach entails using an effective distance measure between samples, but this is difficult to determine, and usually induces some bias within the produced set, which is why the wrapper strategy is adopted in *Blue**. Its authors initially created *RPNI** [27], by modifying the merge operator so as to make it statistically robust to misclassified examples, and later adapted the same approach to EDSM, adding what they called "statistical promotion". Their approach is based on the comparison of misclassified sample proportions, and aims at verifying that such proportions do not increase significantly after a state merging; the resulting reduction in the size of the produced DFA is accepted when the error does not exceed some chosen threshold. Since the error variations might depend on the particular sampling of target language, a simple comparison of misclassified proportion is not sufficient and a statistical method is necessary to deal with error variability. The proposed rule considers a merge as *statistically acceptable* "if and only if the proportion of misclassified examples in a DFA after a merging is not significantly higher then the proportion computed before the merging" [25]. Many tests are available in the context of statistical inference in order to assess two proportions; among them, *hypothesis testing* [28] appears a good candidate for the problem discussed here.

3.2. A Metric accounting for Similarity in Structure and Language

The definition of a reliable similarity measure is one of the most important issues in mobility data analysis, especially since a common area of application falls within the context of recommender systems, where service customization is usually provided depending on

[4] A sample set is said to be structurally complete w.r.t. an automaton, if every transition of the automaton is used by at least a string in the set, and every final state corresponds to at least one string in that set.

DRAFT

the similarities among users. Most metrics documented in literature are heavily based on statistical properties computed on a relevant set of features of the underlying models; however, identifying such features, and devising an effective measure for describing their dissimilarities may be tricky.

In our specific context, we are not merely interested in comparing the structures of the obtained models, but also in capturing the language of the mobility paths, or, in other words, in computing the similarity between mobility habits of users.

We chose to refer to the similarity measure described in [29], whose authors propose to perform the comparison between two models by considering their behavioral aspects; our choice was due to the fact that such measure appears well suited to provide a deep comprehension of automata as language recognizers. Considering two automata, one of them is designated as "target", and the other as "subject"; the aim is to assess how similar the former is to the latter. The main underlying idea is the identification of a significant set of strings to be used as probes, in order to assess how both automata behave when processing them; the similarity score will depend on how many strings are identically classified by both the target and the subject.

The choice of the set of probes, however, is not trivial, because it should (1) encode every reachable state, (2) trigger every transition and (3) preserve the correct arrangement of states. The W-method [30] represents a possible solution; it is based on two sets: the *cover set*, i.e. the set of strings guaranteeing that every state is reached at least once, and the *characterization set*, that ensures that every possible sequence of symbols starts from each state, and furthermore that every unique state from the reference automaton is explored in the subject machine. The *probe set* is obtained as the cross product of the *cover set* and *characterization set*.

However, merely computing a ratio of strings treated identically by both automata would likely result into an unreliable similarity score, which might be biased due to a significant asymmetry between the amount of accepted and rejected examples from the probe set (a common situation in application scenarios for the W-method).

To address this issue, the generated sequences are fed to the DFAs, and the outcome of their classifications are categorized in a confusion matrix, and the similarity measure is computed by means of the F-measure [31], defined as $F = 2 * Precision * Recall/(Precision + Recall)$, which corresponds to computing the harmonic mean between two classic measures of statistical relevance, namely Precision and Recall. Similarity computed in this way naturally emphasizes the importance of capturing the language of the reference machine, rather than ensuring accuracy with respect to language complements.

Finally, our similarity measure S was defined as: $S(A,B) = (S_w(A,B) + S_w(B,A))/2$, where A, B are two automata and S_w is the similarity calculated according to [29], whose first argument is the target automaton and the second is the subject.

4. Representing Mobility Patterns through Regular Grammars

Following the discussion provided in the previous sections, the first step of our approach consists in translating user paths into a symbolic representation; to this aim, we selected an encoding system for geographical coordinates known as *geohash*, which assigns a hash string to each *(latitude, longitude)* pair [32]. The encoding is based on a hierar-

DRAFT

Figure 2. An example of how the first two bits of a geohash string are generated.

chical spatial data structure that recursively subdivides the whole globe into "buckets" according to a grid; unlike traditional coordinate systems, it does not actually represent a point, but rather a bounding area to which the point is restricted. The space is partitioned according to a 4×8 grid; each cell can be recursively divided into 32 smaller cells, and so on, thus providing a hierarchical structure that resembles that of a recursive quadtree; at each iteration, each cell is identified by an alphanumerical character from an alphabet of 32 symbols. This process can be iterated until the desired spatial accuracy is obtained: the longer the geohash string, the smaller the area; an example of encoding at the first 2 levels is reported in Figure 2.

The source data we will consider consists of *movement tracks* [33], i.e. temporally ordered sequences of spatial-temporal position records captured by a device during the whole lifespan of the user observation. Those have to be turned into *trajectories* [34] in order to be able to filter out noise, and to estimate other movement features, such as speed and direction. The true aim of the analysis may however be identified in the *paths*, which are defined as the portion of a trajectory between two relevant points in time or space. Paths reveal user behavior, and highlight relevant places where users spend most of their time. Being aware of these places is crucial in many applications, and they are fundamental in comparing habits of several users or in recognizing anomalies or changes in their routines.

In our approach, trajectories are coded into symbolic sequences by turning each pair of coordinates into the corresponding geohash string; through this encoding, they can easily be analyzed at different spatial scale: once the required precision is set, it is sufficient to truncate every geohash string of each trajectory at the corresponding length. The user mobility model is finally decomposed by following the trajectories with respect to every cell of geohash encoding: a regular language is thus learned for each cell of the geographical area crossed by user movements, starting at the highest level of granularity, as shown in Figure 3. At any level, a more complex and detailed automaton may be obtained by substituting to each symbol the recognizer for the corresponding cell; this is equivalent to concatenating a new symbol to the geohash string, and inspecting the movements at a finer detail. The process stops at the cell granularity representing the required accuracy.

At smaller scales, *mini trajectories* can be obtained for each cell by considering all the contiguous subsequences of strings within each trajectory that share the prefix corresponding to the cell. For each element of the subsequence, only the symbol of the sub-cell is considered, thus the subsequence is turned into a string; after recovering all the strings related to the cell, the needed information to infer a regular language is obtained.

As discussed earlier, a presentation from an informant is required to infer a regular language; so, in order to obtain the mobility model for a user, a set of examples of his paths is not enough. For our case study, we consider the symmetric difference between the set of trajectories of other users and the trajectories of the current user as the negative

DRAFT

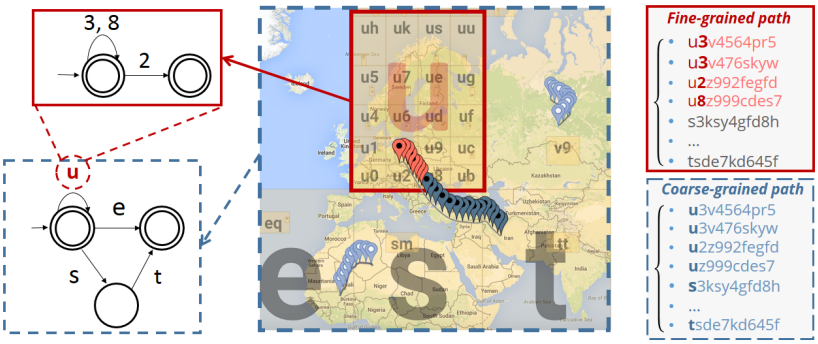


Figure 3. Given the DFA for a larger cell (dashed-line box), a more detailed model can be built by inferring the DFA for transition u (solid-line box).

Table 1. Similarity between original users and artificial one at three selected scales.

Prefix	User				
	3	4	17	30	62
wx	0.37 *(0.008)*	0.34 *(0.001)*	0.41 *(0.007)*	0.39 *(0.160)*	0.78 *(0.001)*
wx4	0.38 *(0.001)*	0.07 *(0.002)*	0.43 *(0.036)*	0.44 *(0.034)*	0.44 *(0.034)*
wx4g	0.44 *(0.040)*	0.48 *(0.001)*	0.52 *(0.001)*	0.40 *(0.014)*	0.37 *(0.003)*

sample set. This set represents viable routes chosen by other users, which have not been traversed by the current user, and can arguably be considered as negative samples for the language representing the mobility habits of the current user. We thus use the *Blue** algorithm to infer the corresponding regular language, given the mini-trajectory sets of negative and positive route samples.

5. Experimental Assessment

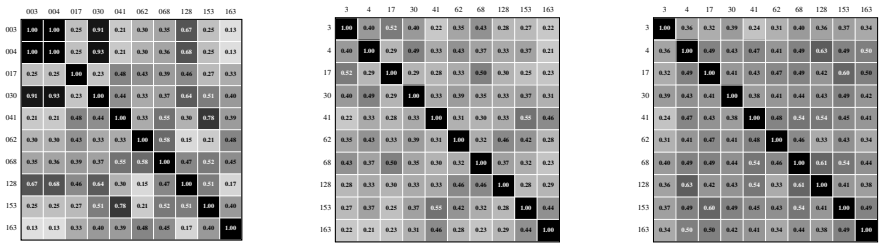
In order to assess our approach, we examined data provided by the *Geolife* dataset [35], which is a collection of time-stamped triples of the form *(latititude, longitude, altitude)*, representing the spatial behavior of 182 users monitored for 5 years, collected by Microsoft Research Asia. Most trajectories took place in China, near Beijing, but routes crossing USA and Europe are also present. More than 17,000 trajectories are contained into the dataset, for a total of approximately 50,000 hours of tracked movements. GPS loggers and smartphones acted as acquisition devices, providing a high sampling rate ($1 \sim 5$ seconds in time, and $5 \sim 10$ meters in space) for more than 90% of the data.

A library, named *GI-learning*, was developed and published[5] for the grammatical inference task. It provides many state-of-the-art algorithms for grammatical inference implemented in C++, and optimized to deal with large quantity of data.

We compare our approach to that proposed in [14], whose performance was tested on the Geolife dataset, so it enables for a direct comparison between results achieved by our system and theirs.

[5] `https://github.com/piecot/GI-learning`

DRAFT



(a) Geohash cell *wx*. (b) Geohash cell *wx4*. (c) Geohash cell *wx4g*.

Figure 4. Similarity matrix for a subset of Geolife users at three different scales.

DRAFT

In a first test, we evaluate the reliability of our similarity measure. However, assessing the effectiveness of a similarity measure is not easy in the field of mobility data, due to the lack of a proper ground truth. We chose to overcome this difficulty by following the same strategy adopted in [14]. The basic idea is to compare the model of a user, with the model of an artificially generated one, obtained by considering only a subset of the original data. In our case, additional, artificial users were generated by randomly selecting half of the paths from five existing users. The similarity measures between an original user and each of its artificially generated offspring were computed, in order to achieve a non-biased estimate. Table 1 reports the mean and variance of the obtained accuracies. Notably, our measure appears reasonable as it captures the similarity between any of the original users and the respective offspring, if compared with those in [14], capturing the same trends. Moreover, the results show that the computed similarity is coherent with data at all the relevant scales.

A further set of experiments was carried out, and the similarity for every pair of users, selected among the ones with the highest amount of paths within Geolife, was computed and compared to the results presented in [14]. A partial report of the results for the second test is shown in Figure 4, where the similarity for 10 users is depicted, at three different scales, with geohash length encoding of 3, 4 and 5, respectively. The results obtained are in accordance to the ones presented in [14] for users considered in both works (namely, users 3, 4, 17, 30, 68, 153, 163), thus confirming that the reliability of the proposed similarity is comparable to the state-of-the-art metrics in literature.

Finally, we would like to point out a peculiarity of our approach in that it allows to represent similarity more expressively than just the figure of its measure. Figure 5, for instance, shows the distribution of the similarity between two users (namely, users 17 and 153) in selected geohash cells. Both users moved within the Beijing area showing an overall similarity of 0.33; thanks to the multi-scale nature of our models, and consequently of our metric, we are able to easily show how the habits of those user compare across different areas of the city. For instance, they appear to behave much more similarly to each other in the upper right corner cell, while they show a much more different behavior in the lower left corner.

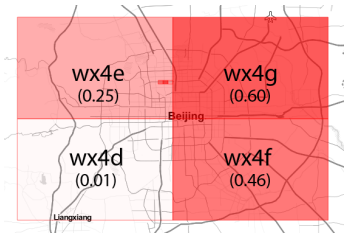


Figure 5. Similarity of two users in the different areas of the same geohash cell.

6. Conclusion

This paper described a structural approach to extract and compare mobility models from huge datasets of positioning data. The main idea is that good quality models can be obtained by coding structures inferred from the wealth of collected samples by a syntactic approach, taking advantage of the natural recursive nature of human mobility habits.

Grammatical Inference is used in order to build models, able to to perform multi-scale analysis and suitable to identify the most relevant relations at different granularities. Moreover, this representation enables a metric for comparing mobility habits that is based on state-of-the-art similarity measure for regular languages. The presented results demonstrate that the proposed multi-scale models and metric perform well on a large dataset of real data, showing promising outcomes and motivating future research. Specifically, we plan to incorporate semantic information about locations to improve the similarity measure and the obtained models.

DRAFT

References

[1] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[2] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, January 2006.

[3] Roberto Trasarti, Fabio Pinelli, Mirco Nanni, and Fosca Giannotti. Mining mobility user profiles for car pooling. In *Proc. of the 17th ACM Int. Conf. on Knowledge discovery and data mining*, pages 1190–1198. ACM, 2011.

[4] Tim Oates, Arnold P. Boedihardjo, Jessica Lin, Crystal Chen, Susan Frankenstein, and Sunil Gandhi. Motif discovery in spatial trajectories using grammar inference. In *Proceedings of the 22th ACM Int. Conf. on Information & Knowledge Management*, pages 1465–1468, New York, NY, USA, 2013. ACM.

[5] S. C. Geyik, E. Bulut, and B. K. Szymanski. Grammatical inference for modeling mobility patterns in networks. *IEEE Transactions on Mobile Computing*, 12(11):2119–2131, Nov 2013.

[6] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Lszl Barabsi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[7] Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, and Marco Ortolani. A machine learning approach for user localization exploiting connectivity data. *Engineering Applications of Artificial Intelligence*, 50:125 – 134, 2016.

[8] S. Gaglio, G. Lo Re, and M. Morana. Real-time detection of twitter social events from the user's perspective. In *IEEE International Conference on Communications (ICC2015)*, pages 2810–2815, 2015.

[9] Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, and Marco Ortolani. User activity recognition for energy saving in smart homes. *Pervasive and Mobile Computing*, 16, Part A:156 – 170, 2015.

[10] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 1082–1090. ACM, 2011.

[11] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Transaction on the Web*, February 2011.

[12] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proce. of the 18th Int. Conf. on Advances in Geographic Information Systems*, New York, NY, USA, 2010. ACM.

[13] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, GIS '11, pages 34–43, New York, NY, USA, 2011. ACM.

[14] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles. *ACM Trans. Web*, 8(4):21:1–21:25, November 2014.

[15] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2007.

[16] Mingqi Lv, Ling Chen, and Gencai Chen. Mining user similarity based on routine activities. *Information Sciences*, 236:17 – 32, 2013.

[17] Zoltn Koppnyi. Individual human mobility modeling with probabilistic automata. In *Proceedings of the Conference of Junior Researchers in Civil Engineering*, 2012.

[18] Alessandra De Paola, Salvatore Gaglio, Giuseppe Lo Re, and Marco Ortolani. Multi-sensor fusion through adaptive bayesian networks. In *AI*IA 2011: Artificial Intelligence Around Man and Beyond*, volume 6934 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2011.

[19] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[20] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[21] Colin de la Higuera. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

[22] P. Dupont, L. Miclet, and E. Vidal. What is the Search Space of the Regular Inference? In *In Proc. of the Second Int. Coll. on Grammatical Inference (ICGI'94)*, pages 25–37. Springer Verlag, 1994.

[23] José Oncina and Pedro García. Identifying regular languages in polynomial time. *Advances in Structural and Syntactic Pattern Recognition*, 5(99-108):15–20, 1992.

[24] Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo One DFA learning competition and a new Evidence-Driven State Merging Algorithm. In *Proc. of the 4th Int. Coll. on Grammatical Inference (ICGI '98)*, pages 1–12. Springer-Verlag, 1998.

[25] Marc Sebban, Jean-Christophe Janodet, and Frédéric Tantini. Blue: a blue-fringe procedure for learning dfa with noisy data.

[26] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(12):273 – 324, 1997.

[27] Marc Sebban and Jean-Christophe Janodet. On state merging in grammatical inference: A statistical approach for dealing with noisy data. In *Machine Learning, Proceedings of the Twentieth International Conference USA*, 2003.

[28] Ken Black. *Business statistics: for contemporary decision making*. John Wiley & Sons, 2011.

[29] Neil Walkinshaw and Kirill Bogdanov. Automated comparison of state-based software models in terms of their language and structure. *ACM Trans. Softw. Eng. Methodol.*, 22(2):13:1–13:37, March 2013.

[30] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Trans. Softw. Eng.*, 4(3):178–187, May 1978.

[31] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.*, 45(4):427–437, July 2009.

[32] Zoran Balkić, Damir Šoštarić, and Goran Horvat. Geohash and uuid identifier for multi-agent systems. In *Proceedings of the 6th KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications*, KES-AMSTA'12, pages 290–298. Springer-Verlag, Berlin, Heidelberg, 2012.

[33] Chiara Renso, Stefano Spaccapietra, and Esteban Zimnyi. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press, 2013.

[34] Yu Zheng and Xiaofang Zhou. *Computing with Spatial Trajectories*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[35] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proc. of the 17th Int. Conf. on World Wide Web*, pages 247–256, New York, NY, USA, 2008. ACM.

DRAFT