ECAI 2016
G.A. Kaminka et al. (Eds.)

© 2016 The Authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/978-1-61499-672-9-999

Gaining Insight by Structural Knowledge Extraction

Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani[1]

Abstract. The availability of increasingly larger and more complex datasets has boosted the demand for systems able to analyze them automatically. The design and implementation of effective systems requires coding knowledge about the application domain inside the system itself; however, the designer is expected to intuitively grasp the most relevant features of the raw data as a preliminary step.

In this paper we propose a framework to get useful insight about a set of complex data, and we claim that a shift in perspective may be of help to tackle with the unaddressed goal of representing knowledge by means of the structure inferred from the collected samples. We will present a formulation of knowledge extraction in terms of Grammatical Inference (GI), an inductive process able to select the best grammar consistent with the samples, and a proof-of-concept application in a scenario of mobility data.

1 Introduction

Knowledge extraction has represented one of the most interesting challenges in Artificial Intelligence for the past decades [1]. Massive collections of data regarding the most disparate aspects of users' lives have become readily available for machine processing, deeply changing the nature of the problem. Nowadays the main concern is not just the necessity of accurate predictive models, but above all the demand for early provision of reliable insights to experts. The main issue regards the choice of the most appropriate tools and features to extract information from high-dimensional, incomplete and noisy datasets. Researchers have become increasingly more aware that "measuring" does not seamlessly translate into "understanding", and their primary goal is to make sense of data by letting models *emerge* from the collected samples, rather then deducing them from pre-set assumptions. In this context, an essential requirement is the ability to build models that may be interrogated in order to improve representation and comprehension about the nature of data. A vast literature has investigated the interpretability of models and results produced by learning algorithms: approaches of increasing complexity have provided more and more accurate results, at the cost of less transparent representations [2]. Often, predictions supplied by these methods help the user in choosing the best option among several available ones; without interpretable models, this process can not provide any remarkable insight to support and explain the decision. We claim that an approach to knowledge extraction that highlights the structural information can alleviate this problem. Specifically, we propose to represent the meaningful information by means of the *structure* inferred from the collected samples. Our definition of structural knowledge refers to the taxonomy proposed in [3], where three different types of knowledge are singled out:

- *declarative* knowledge expresses the awareness about some items, events or concepts. It is the knowledge about *"knowing that"*, which allows us to identify and describe an item or a concept, but does not enable us to use them;
- *procedural* knowledge describes how learners use or apply the former type of knowledge; it is about *"knowing how"* to do something.
- *structural* knowledge mediates the translation of declarative into procedural knowledge and facilitates the application of the latter; it refers to how concepts within a domain are interrelated; it is the knowledge about *"knowing why"*.

We note that *structural* knowledge is significantly different from *structured* knowledge, in that the latter typically refers to a description through entities and relationships; in other words, the focus is on how knowledge itself is organized. On the other hand, structural knowledge deals with the type of knowledge to be acquired, rather than the way it is organized. The emphasis is on the organization and structure of the objects of the analysis, and this will be the topic of our discussion.

In the present paper, the process of automatic extraction of this type of knowledge from raw data will rely on concepts and methods from *Algorithmic Learning Theory* (ALT), whose main subject is the study of formal languages and automata. Unlike its statistical counterpart, ALT does not require any specific constraints on the statistic properties of the available data, so it is well suited for cases when no a-priori hypotheses can be formulated. Its most interesting peculiarity is that the obtained knowledge is syntactically driven, hence intrinsically structural. Thus, representations obtained through algorithmic approaches can point out interesting relationships among the key elements of a dataset, implicitly suggesting what the most relevant features are. In particular, we will make use of *Grammatical Inference* (GI) [4], an inductive process able to select the best grammar (according to a metric) that is consistent with the samples. Instead of being represented in a vectorial space, we will thus regard our input as strings generated by an unknown grammar [5]; our claim is that GI can be successfully applied in order to get relevant insights about the hidden structure embedded in large collections of

[1] DICGIM – University of Palermo, Italy, email: *first-name.lastname*@unipa.it

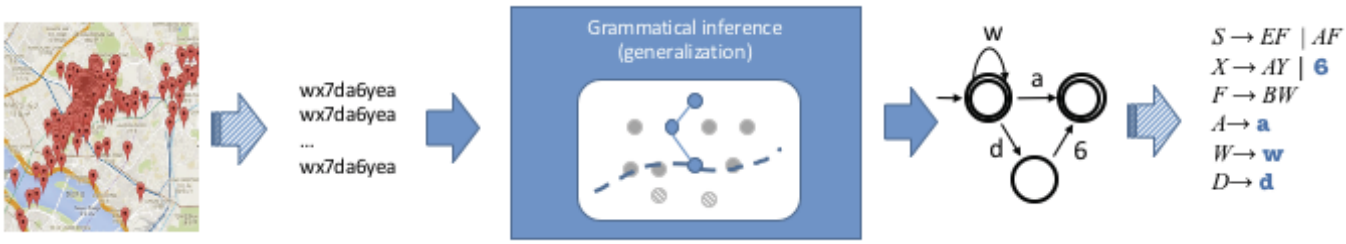


Figure 1. From data to grammars: an overview of the proposed approach.

data, enabling the user to pose new kinds of questions, taking advantage of the generative models obtained by the inductive process. Thanks to their recursive nature, grammars are also able to make recurrent relations among data explicit at different granularities.

In order to highlight the potential of the suggested approach, GI, and more specifically inference of regular languages [6], has been applied to the problem of inferring mobility models. In this context, multi-scale analysis allows us to grasp a more significant insight into data, and to get a better representation of user mobility habits, according to the traversed routes.

The remainder of the paper is structured as follows. Section 2 contains a very brief survey of methods for coping with high-dimensional and complex datasets. In Section 3 our approach based on GI will be described, followed by a case-study application to mobility data in Section 4. Finally, we will present our conclusions in Section 5.

2 Preliminaries: coping with dimensionality

Learning from experience is a key point in the design of intelligent agents. Over the years, this issue has been addressed in different ways, depending on the available devices, algorithms, and data, beginning with expert systems, probabilistic graphical models, and other statistical approaches. It soon became apparent, however, that one of the most relevant challenges was the selection of features from unlabeled data, so a lot of effort has been devoted during the last decade to the creation of systems able to perform this task automatically. Notable examples of this class of methods fall under the name of *Deep Learning*, and it has been shown that their finest performance is comparable to the best hand-engineered systems [7], [8]. A strong theoretical limitation, however, is represented by the well-known *No Free Lunch* theorem; one of its formulations informally states that "any two optimization algorithms are equivalent when their performance is averaged across all possible problems" [9]; in other words, there is no possible general criterion for choosing the optimal parameters of a method when absolutely no prior knowledge about the problem is available, except raw data [10]. If models are to be regarded as "black-boxes", there is no reasonably efficient way to choose among several of them, when all choices fit the data comparably well.

The most recent technological advances have once more complicated the nature of the problem; it is now possible to perform measurements regarding the most disparate aspects of users' lives at previously inconceivable rates; moreover such data are highly heterogeneous, so the obtained datasets are typically high-dimensional and possibly incomplete. One of the most common examples is the massive volume of data with diverse features collected in smart environments [11], where pervasive networks of sensing devices are deployed, in order to support users in controlling the monitored environments [12], [13]. The peculiar challenges related to the analysis of this kind of data has given rise to a specific branch of AI named *Ambient Intelligence* (AmI), specifically aimed at exploiting the information about the environment state in order to personalize it, adapting the environment to users' preferences [14].

In this context, most traditional approaches to data mining are not viable to handle the complexity of the new collections especially because they fail to provide useful insight into the real nature of data [15]. Very high dimensionality is hardly manageable by a human mind so, lacking support from the machine, designers are effectively prevented from grasping the most important features to consider.

It has thus been claimed [16] that the availability of *qualitative information* might ease the problem: at the cost of decreasing accuracy, the user can obtain a better understanding of the data, being free to focus on the overall organization at a larger scale; once a first insight is obtained, the process can be repeated at a smaller scale, considering only a subset of the original dataset, or a projection with lower dimensionality.

In this paper, we claim that qualitative information can provide very useful and compact guidelines to designers, in the preliminary set-up of systems for automatic data analysis. Also, recent findings [17] show that neural processes activated by human comprehension hint toward a grammar-based inner construction of knowledge representation; hence, modeling data in the form of grammars might help users to figure out the main structure behind relevant information. Grammar representations have been devised for *syntactic pattern recognition* [18]; in this work, we use some ideas pertaining to this research area, adapting and updating them with recent advances in data analysis.

3 Inferring the structure behind data through formal grammars

Assuming as a working hypothesis that the environment observed by the agent is computable, our goal is to exploit the

available data in order to infer a model that closely matches the unknown model for the environment. In other words, we assume that a (yet unknown) language describing our data exists; admittedly, this language may be extremely complex and data may be corrupted by noise, so that reconstructing the original language from raw data is likely to be a very challenging task. However, relying on formal languages to represent, organize and process knowledge is advantageous as they naturally provide a description of the relations between their elements, which may be regarded as their *hidden structure*.

A formal language is a (finite or infinite) set of sentences, each finite in length and made up of a finite set of symbols [19]. In real-life problems, however, data is often represented by a projection in a geometric space, whose dimensions are the chosen features, so a preliminary step requires translating the original representation of the data into a symbolic one. This is the first step shown in Figure 1, which depicts a high-level representation of our approach.

By encoding data as symbolic strings, we in fact move from a representation in a classical Euclidean space to a hierarchical organization; we rely on an ultra-metric space organized as a tree, where each node is associated to a string representing its path from the root, as will be detailed in the next sections.

The core of our approach is to use the symbolic data to infer the underlying target language through one of its possible representations. Generally speaking, two different descriptions can be associated to a language, namely a *generative* description, and a *recognition-based* one. In this paper, we focus on regular languages, so the corresponding representations are *regular grammars* and *Deterministic Finite Automata* (DFAs), respectively.

The *generative* description corresponds to a grammar, that is a formal system able to produce exactly the set of strings of the given language by applying predefined rewriting rules, expressed in the form of productions [20], [21]. A taxonomy of grammars has been proposed based on the complexity of such transformation rules, with regular grammars at its lowest level [22]. Generative descriptions are appealing to humans because they are intuitive, but their straightforward implementation is inefficient.

In the *recognition-based* description, a language is considered as the set of strings accepted by an automaton, that is a formal system that accepts all the set of strings belonging to the given language rejects the others. Automata are appealing to machines, because they are formal, compact, low-level machines and can be implemented easily and efficiently; on the other hand, they are hardly understandable by a user.

Inferring a language through a grammar is by all means a learning process which may be characterized by its capability of *generalizing*. Unlike other learning approaches, where generalization is obtained by optimization, GI belongs to the category of algorithms that generalize through a search in a hypothesis space, so it may be regarded as an instance of the general framework known as *Version Space* strategy [23]. The key insight of this strategy is the assumption that hypotheses in the search space are organized through a "general-to-specific" ordering; a learning algorithm can explore the infinite hypothesis space by exploiting its structure, without explicitly visiting every element of it. In GI, the general-to-specific ordering is defined in terms of relations between automata, and the order is thus induced on languages.

3.1 Grammatical inference

As stated in [4], identifying a language is the main concern of *Grammatical Inference* (GI), which may be defined as the process of searching for a hidden grammar by exploiting the scarce available information, often consisting of just a set of strings; as such, GI belongs to the broader framework of *Algorithmic Learning Theory* (ALT), whose central concept is that of a *language learnability model*. Its main components are a *definition of learnability*, a *method of information presentation*, and a *naming relation*.

In this context, learnability is expressed by the principle of *identification in the limit* formulated by Gold [24]: the learning algorithm should identify the correct hypothesis on every possible data sequence consistent with the problem space. This idea is a non-probabilistic equivalent of statistical consistency, where the learner can fail on data sequences whose probability measure is 0; in this case, a learner (an algorithm) will identify a language in the limit if, after a number of presented strings, *its hypothesis no longer changes*.

The way in which input data are provided to the learner is called a *presentation*; let L indicate a language defined over an alphabet Σ, this is a function $\phi : \mathbb{N} \to X$, defined over the set of natural numbers, with codomain some set of samples $X \subset L$. As regards the methods of information presentation, two main procedures are available:

– presentation from *text*: a sequence of strings $(x_1,\ x_2,\ \ldots)$ belonging to L is provided; every string in L must appear at least once in the sequence. This presentation, denoted by $T(L)$, is also known as *positive* presentation:

$$T(L) = \{\phi : \mathbb{N} \to \Sigma^* : \phi(\mathbb{N}) = L\};$$

– presentation from *informant*: the learner is supplied with strings marked as *positve* (i.e. belonging to the language L) or *negative* (not in L). This kind of presentation, denoted by $I(L)$, is known as *complete*:

$$I(L) = \{\phi : \mathbb{N} \to \Sigma^* \times \{0,\ 1\} : \phi(\mathbb{N}) = L \times \{1\} \cup \overline{L} \times \{0\}\},$$

where $\overline{L}$ indicates the complement of L with respect to Σ^*.

Finally, the naming function is some surjective function $\mathbb{L} : \boldsymbol{\mathcal{G}} \to \boldsymbol{\mathcal{L}}$, with the set $\boldsymbol{\mathcal{G}}$ of grammars as the domain, and the set of languages $\boldsymbol{\mathcal{L}}$ as the codomain.

The language learnability paradigm has some theoretical limitations. As Gold showed in [24], a class of *super-finite languages*[2] cannot be identified in the limit from a text presentation. This class includes regular languages, hence they cannot be inferred from positive examples only; in other words, a set of strings belonging to the target regular language is not sufficient to learn it.

Even if we turn to a presentation from informant, we incur some limitations, as also pointed out in [24]. In particular, the following holds:

Theorem 1 *The whole class of recursive languages can not be identified in the limit from a complete presentation.*

However in the same work Gold showed that, when we restrict the class of languages, it may be proven that:

[2] A super-finite language class is a class that contains all finite languages and at least one infinite language.

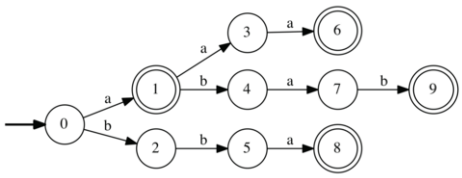


Figure 2. $PTA(I_+)$ for $I_+ = \{a, aaa, abab, bba\}$.

Theorem 2 *The class of primitive recursive languages*[3] *can be identified in the limit by a complete presentation.*

Regular languages are primitive recursive languages, so a complete presentation of examples guarantees their theoretical learnability. We can thus turn our attention to how the inference process can be practically carried out. Motivated by the generalization principle, we are interested in identifying the most general DFA consistent with the given samples, i.e. the minimum canonical automaton.

Even though, given a complete presentation of positive and negative examples $I = I_+ \cup I_-$, an automaton consistent with I exists and is unique [20], Gold also showed that finding the minimum consistent automaton with a set of samples is an NP-hard problem; therefore, some heuristic is needed to carry out this search in an efficient way.

3.2 Generalization as search

We will characterize the search space for our problem through the following basic elements:
- *initial node*: an "acceptable" DFA;
- *successor function*: a set of successors of an automaton generated by pairwise state merging;
- *target*: minimum automaton that is consistent with the samples I.

This search space may thus be described as a Boolean lattice [26], whose initial node is a tree automaton – the so-called *Prefix Tree Acceptor* (PTA) – accepting precisely the positive examples I_+, such as the one shown in Figure 2.

The complexity of the search can be eased by exploiting some general-to-specific ordering of the nodes; intuitively, in grammatical induction, this ordering is based on constraints characterizing the hypotheses, with fewer constraints entailing more general hypotheses, and vice versa. By construction, the PTA(I_+) is the most specific DFA for the positive examples, and we want to explore the space moving toward the minimum consistent automaton, with the negative examples as our bounds.

The set of successors of an automaton is generated by pairwise merging operations: two states of the original automaton are chosen for merging, resulting into a new automaton with one fewer state with respect to the original, as shown in Figure 3 which depicts an excerpt of a lattice. Even though merging two states might result into a non-deterministic automaton, it is possible to carry out the generalization process

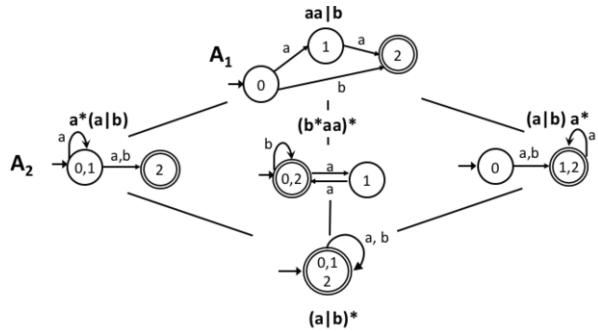


Figure 3. Excerpt of a lattice: automata in the middle row are obtained from the initial one by merging different pairs of its states.

avoiding non-determinism, by making use of the so-called *folding* operation, as described in [4].

Pairwise merging may be formally defined as a partition of the set of states of the original automaton A, and is a *derivation operation*, which defines a partial order relation over the set $\Pi(A)$ of all the possible partitions of the set of states in A. Notably, it preserves the property of *language inclusion*, as shown in [26], which means that the application of the merging operator:
- either causes the number of states to decrease, but the recognized language is preserved;
- or it also implies a change in the language recognized by the resulting automaton, but such language is more general, and properly includes the original one.

The Boolean lattice $Lat(PTA(I_+))$ is thus completely defined by its initial node, i.e. $PTA(I_+)$, and the nodes obtained by repeatedly applying merging operations included in $\Pi(PTA(I_+))$; the deepest node in $Lat(PTA(I_+))$ is the *Universal Automaton* (UA), that accepts all the strings defined over an alphabet Σ. The inference of regular languages, provided a presentation from an informant, can be turned into the search for an automaton $A' \in Lat(PTA(I_+))$, given the additional hypothesis of structural completeness of I_+[4].

It may be proven [26] that if I_+ is a structurally complete sample with respect to the minimal automaton A accepting a regular language L, then A belongs to $Lat(PTA(I_+))$, so the inference of a regular language by presentation from an informant can be turned into the search for an automaton in the space defined by that Boolean lattice.

The definition of minimal DFA consistent with the sample set I can also be visualized in terms of the elements of the lattice, through the so-called *Border Set*, which establishes the limit of generalization in the search process under the control of negative samples I_-, as graphically shown by the dotted line in Figure 4. The border set parts the lattice into two main subsets: *admissible* automata, between the root and the border, and inadmissible ones, falling beyond the border. The minimum DFA consistent with I is the deepest (i.e. smallest) automaton falling right on the border set, hence still admissible.

[3] A language is primitive recursive if its characteristic function is primitive recursive. The formal definition may be found for instance in [25].

[4] A I_+ sample set is said to be structurally complete with respect to an automaton A, if every transition of A is used by at least a string in I_+, and every final state in A corresponds to at least one string in I_+.

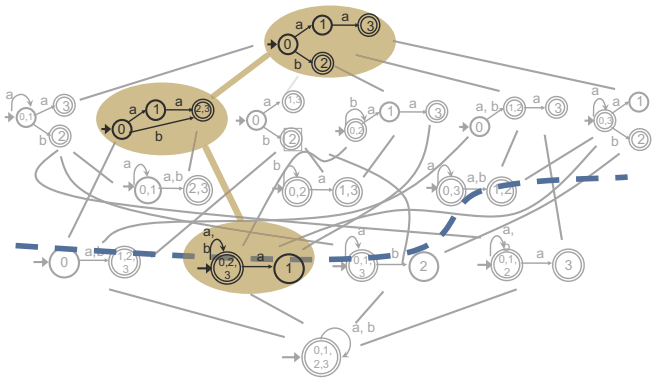


Figure 4. Sketch of a search in the Boolean lattice.

Since the number of automata in the lattice generated by an initial PTA with n states is given by the *Bell number*:

$$\omega(n) = \sum_{p=0}^{n-1} \binom{n-1}{p} \omega(n-1),$$

with $\omega(0) = 1$, then the space defined by such lattice is clearly too large to be searched exhaustively; therefore, some approaches have been proposed to carry out the search more efficiently.

Evidence-Driven State Merging (EDSM) represents a state-of-the-art iterative algorithm to perform such search, whose detailed description can be found in [6]. It was introduced to reduce the number of comparisons necessary during merging, and makes use of a heuristic that computes a score for all possible merges by counting the number of strings that would end in the same state; the function returns $-\infty$ if the merge makes the automaton inadmissible (i.e. an element of I_- would be accepted or an element of I_+ rejected); the pair with the highest score is chosen for merging.

The results of applying this algorithm in order to perform structural knowledge extraction in a practical scenario will be discussed in the next section.

4 A proof of concept: mobility data

In order to provide a proof of concept for our approach to structural knowledge extraction, we consider a case study aimed to infer and represent user mobility models via regular languages.

A mobility model is a concise and meaningful representation of past and future mobility behaviors of users. Nowadays, location data are easy to collect, thanks to the availability of a wide set of common devices, such as smartphones or tablets, that easily provide large amounts of measurements [27]. Extracting meaningful information from this wealth of data, however, is still an open issue. The main questions, for instance, regard the selection of the most significant features, the proper granularity necessary to perform effective analysis, and the metric to use to compare the mobility habits of various users.

Figure 5. An example of the first two bits of a geohash string.

4.1 Positions as symbols

Following the approach presented in the previous sections, the first step of the process requires translating paths into a symbolic representation; to this aim, we selected an encoding system for geographical coordinates known as *geohash* encoding.

Geohash assigns a hash string to each *(latitude, longitude)* pair; originally, it was developed to provide a smart and easy representation of URLs, but it has been since widely used to store spatial coordinates in databases [28]. The encoding is based on a hierarchical 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.

In the geohash string, even bits encode information about longitude, while odd ones encode latitude; an example of encoding at the first 2 levels is reported in Figure 5, which shows two rectangles partitioning the entire globe longitudinally (left), and the four rectangles that may be obtained with a successive latitudinal partition (right). This process can be iterated until the desired spatial accuracy is obtained: the longer the geohash string, the smaller the area. The length of the binary string must always be a multiple of 5 to allow its conversion to a sequence of symbols from geohash alphabet.

The following table shows the size of the area identified by a geohash code with respect to its length.

Table 1. Area covered by a cell with respect to the length of its geohash encoding string.

Geohash length	$\approx$ Covered Area km^2
1	16,000,000
2	500,000
3	15,000
4	500
5	15
6	0.5
7	0.02

Geohash encoding possesses two notable properties, namely:

– *inclusion*: it is always possible to add a character to a geohash string, obtaining a new string that identifies a cell contained in the original one. For example, the coordinates (38.120281, 13.357278) identify a point included inside the `sqc2zg` cell, but also inside `sqc2zgw` or `sqc2zgwk`;

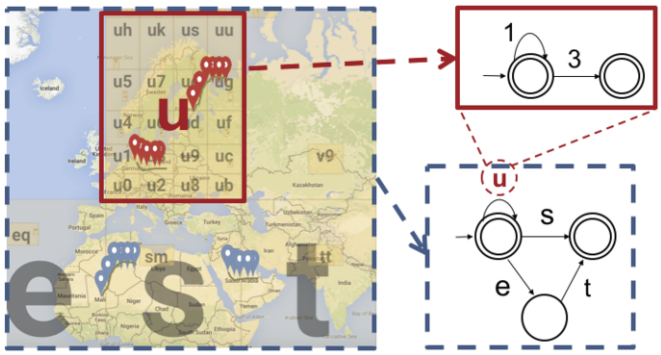


Figure 6. From trajectories to DFA hierarchy: given the DFA for a larger cell (dashed-line box), a more detailed model can be built by inferring the DFA for transition u (solid-line box).

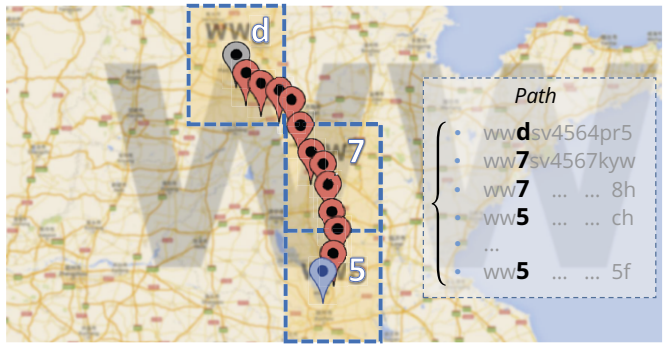


Figure 7. Subsequences with the same prefix originate mini trajectories: the third element of each string (in bold face) may be concatenated to obtain a mini trajectory.

– *locality*: strings with common prefix mark contiguous cells. Thus, it is very simple to check if two cells are neighbors. The converse is not always true: two cells could be next to each other even if they do not share a common prefix.

In the following, we exploit both properties to achieve an effective implementation of our GI process.

4.2 Mobility models as automata

The source data we will consider consists of *movement tracks* [29]:

Definition 1 (Movement track) *This is a temporally ordered sequence of spatial-temporal position records captured by a device during the whole lifespan of the user observation. Each record contains a position and the instant of the capture, with no two records having the same instant value.*

Movement tracks have to be turned into *trajectories* [30] 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*:

Definition 2 (Path) *A path is the portion of a trajectory between two relevant points in time or space dimensions.*

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 transformed 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 scales: 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 6. At any level, a more complex and detailed automaton may be obtained by substituting to each symbol the recognizer for the corresponding cell (see Figure 6); 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 (see Figure 7); 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 models for a user, a set of examples of their paths is not enough. Selecting a proper negative sample set in case only positive samples are available is an open issue in GI. As a practical solution, we consider the symmetric difference between the trajectories of other users and those of the current one, as they intuitively provide valid trajectories that were not actually traversed by the specific user. These samples can be considered as the negative sample set for the language representing the mobility habits of the current user. We thus use the EDSM algorithm to infer the corresponding regular language, given the mini-trajectory sets of negative and positive route samples.

4.3 The language of paths

In order to assess our approach, we examined data provided by the *Geolife* dataset [31], 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 in 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.

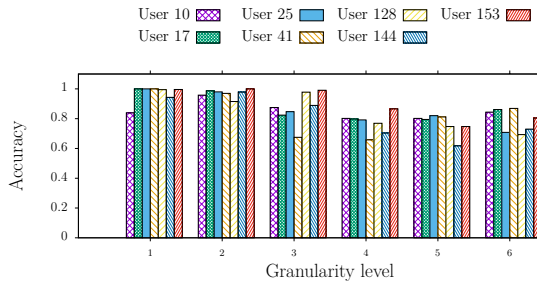


Figure 8. Accuracy with respect to varying granularity for 7 users (80% training, 20% test).

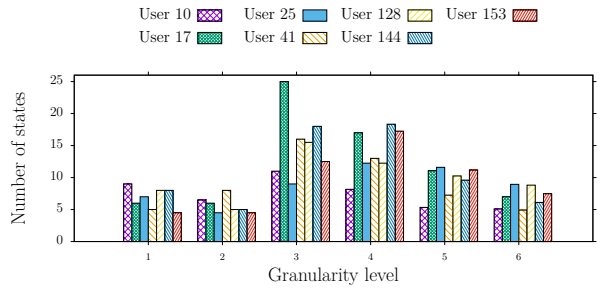


Figure 9. Number of states with respect to the granularity of considered paths.

The goal of our analysis is to address the main issues in the analysis of mobility data, as mentioned at the beginning of this section, and in particular: to assess the correlation between the complexity of spatial behaviors and the respective complexity of the model; to figure out the most representative granularity level that should be considered; and finally to provide a guideline for the definition of a mobility similarity measure between users.

The first issue was addressed by estimating the accuracy of our structural models at varying granularity degrees; in this context, we refer to the classical definition of accuracy as the ratio between the number of corrected classified samples (paths that were actually travelled by the current user, or are correctly disregarded) and the number of the examined samples (all the paths to be classified). We chose to ignore measurements about altitude, as they did not provide any significant information, and to consider only longitude and latitude; no further prior knowledge was assumed. Our analysis was based on the selection of the subset of the most representative users, i.e. the ones that have at least 300 paths at every granularity. For each user, the available data were partitioned into training (80%) and testing (the remaining 20%). The maximum string length for geohash encoding was set to 7, corresponding to a precision of 153 meters (an area of about 0.02 km^2). Results for seven representative users are reported in Figure 8, which shows that high accuracy is obtained at all spatial scales; moreover, it is evident that performances are satisfactory even with higher resolution trajectories. We are thus supported in our claim that the complexity of the users' spatial behavior may be captured by models as simple as regular languages. This is confirmed by several other works in literature [32], [33], [34] that, based on statistical approaches, revealed that human spatial trajectories are highly predictable by simple models: meaningful patterns can be described by a sequence of locations, and are characterized by particular shapes. Thus, we can reasonably conclude that a preliminary insight on data is able to hint the complexity of the model needed for a deeper analysis.

As a second step, we moved on to assess how choosing different granularities affects the complexity of the resulting models, measured in terms of number of states; as is clear from Figure 9, which reports values averaged over all automata for each granularity, medium granularities (encoding lengths 3 and 4) require more complex recognizers with respect to both higher and lower granularities. Arguably, user mobility shows the highest variability at intermediate spatial resolutions (e.g. city-wide), where more features are needed to separate different behaviors, whereas most users typically remain within the same nation, thus exhibiting a simpler behavior that can be explained through a less complex model.

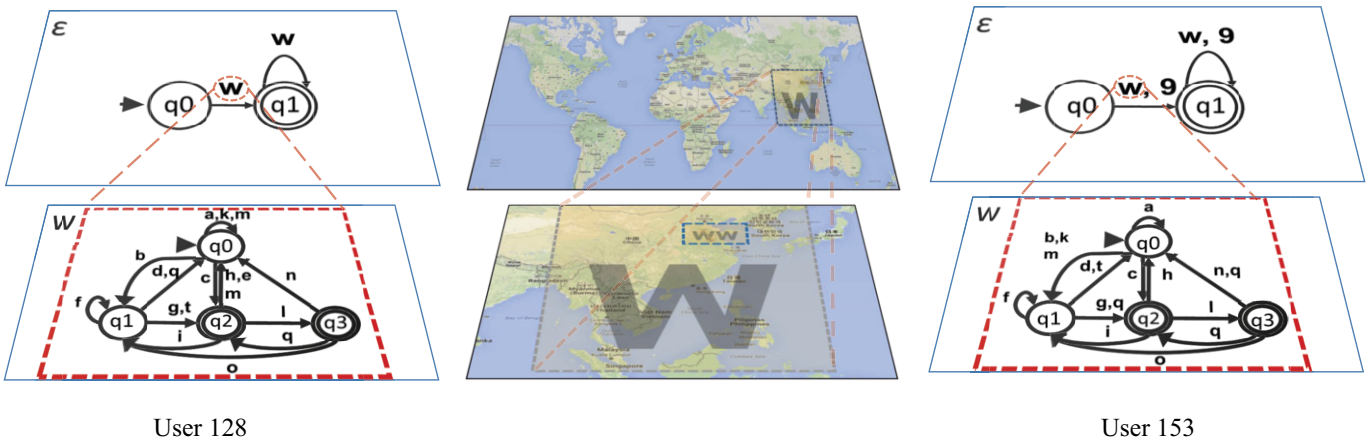
Finally, we tackled the challenge of providing insight about how to identify similarities among users; here, we refer to the definition of similarity used in [35], i.e, a measure for capturing the affinity between two users according to their trajectory patterns, encoded in the respective mobility profiles. Due to the intrinsic recursive nature of users' paths, it is very common that pronounced similarities emerge naturally both at a sufficiently low, and at a very high resolution. In fact, most users share the same behavior at nation-wide scale, since they spend most of their time without leaving their own country; at the other end of the scale, short paths are typically very basic, due to the physical constraints of the urban landscape, so most users will likely show similarities when they traverse small areas, e.g. within a few blocks.

Those considerations appear in all evidence from our experiments, and two representative cases are reported in Figure 10. The first row shows the automata produced for users 128 and 153 at the highest possible granularity, i.e. considering only the first symbol of the geohash encoding. By referring to Table 1 we see that such granularity corresponds to an area of 16 million km^2, such as an entire country or larger; the automaton for user 128 tells us that its strings contain just one symbol (`w`, which is the geohash code roughly covering China), whereas for user 153 two symbols are allowed (`w` and `9`, which roughly encode China and the USA/Mexico region, respectively).

It is informative to look at the alternative representation of the two automata in the form of the corresponding regular grammars:

$$\mathcal{G}_{128}: \begin{cases} S \to WS \mid W \\ W \to \mathbf{w} \end{cases} \qquad \mathcal{G}_{153}: \begin{cases} S \to WS \mid W \\ \quad \mid \ YS \mid Y \\ W \to \mathbf{w} \\ Y \to \mathbf{9} \end{cases}$$

where the dissimilarities between the two users are evident: despite the fact that there exist trajectories for both users that are confined within China, as coded by the W productions, only user 153 moves to a different area altogether, as represented by the Y productions in the rightmost grammar.



User 128 User 153

Figure 10. Comparison of DFAs representing the movements of two users, at different levels of granularity.

We can now exploit one of the key peculiarities of our structural analysis, namely the possibility of a simple navigation across different spatial scales of granularity or, equivalently, a "hierarchical" navigation through the pool of automata obtained by specifying a symbol in a transition through a more specific automaton representing a finer detail (see Figure 10). The automata depicted in the bottom row "expand" the **w**-transitions of the corresponding upper-level automata; in other words, they specify the behavior for each of the two users when they move within the region encoded as **w**[5]. Visually inspecting such automata is sufficient to recognize their similarities; this qualitatively shows that paths have a *multi-scale* nature: significant information can be extracted by observing data at different granularity degrees, and a similarity metric should take this characteristic into account. Our structural models are able to highlight the most appropriate representation level for the problem, hence to provide useful insight to the system designer.

5 Conclusion

This paper described a proposal for a structural approach to coping with the complexity represented by big collections of data. Our claim is that often knowledge can be represented by means of the structure inferred from the wealth of collected samples, limiting the amount of a-priori information needed.

By using a syntactically driven inference algorithm, we showed that it is possible to build generative models able to suggest the relevant relations between different subsets of the samples, and to perform multi-scale analysis suitable to identify the most important features emerging at different granularities.

The presented results, regarding the issue of understanding mobility data, show how, in this context, the availability of generative and multi-scale models allows to get a useful insight of the whole dataset.

[5] For instance, a valid string for the automaton at the lower left is **akc**; this means that the user is moving across cells whose geohash codes are **a**, **k**, and **c**, all of which are subcells of macro-region **w**.

Acknowledgments

The authors would like to thank Gabriele Pergola, student at our lab, for his enthusiastic help during the implementation of parts of the system, and for proofreading the article.

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.

References

[1] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–54, 1996.

[2] Z. C. Lipton, "The mythos of model interpretability," in *Proceedings of the workshop on human interpretability in machine learning*, ser. WHI 2016, New York, NY, USA, 2016.

[3] D. H. Jonassen, K. Beissner, and M. Yacci, *Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge*. Psychology Press, 1993.

[4] C. de la Higuera, *Grammatical inference: Learning automata and grammars*. New York, NY, USA: Cambridge University Press, 2010.

[5] D. Angluin, "Queries and concept learning," *Machine learning*, vol. 2, no. 4, pp. 319–342, Apr. 1988.

[6] K. J. Lang, B. A. Pearlmutter, and R. A. Price, "Results of the Abbadingo One DFA learning competition and a new Evidence-Driven State Merging Algorithm," in *Proceedings of the 4th international colloquium on grammatical inference*, ser. ICGI '98, London, UK: Springer-Verlag, 1998, pp. 1–12.

[7] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in machine learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on machine learning*, ser. ICML '08, Helsinki, Finland: ACM, 2008, pp. 1096–1103.

[9] D. H. Wolpert and W. G. Macready, "Coevolutionary free lunches," *IEEE transactions on evolutionary computation*, vol. 9, no. 6, pp. 721–735, Dec. 2005.

[10] D. Whitley and J. P. Watson, "Complexity theory and the no free lunch theorem," in *Search methodologies: Introductory tutorials in optimization and decision support techniques*, E. K. Burke and G. Kendall, Eds. Boston, MA: Springer US, 2005, pp. 317–339.

[11] A. R. Ganguly, J. Gama, O. A. Omitaomu, M. M. Gaber, and R. R. Vatsavai, *Knowledge discovery from sensor data*, 1st. Boca Raton, FL, USA: CRC Press, Inc., 2008.

[12] P. Cottone, S. Gaglio, G. Lo Re, and M. Ortolani, "User activity recognition for energy saving in smart homes," *Pervasive and mobile computing*, vol. 16, Part A, pp. 156–170, 2015.

[13] S. Gaglio, G. Lo Re, and M. Morana, "Human activity recognition process using 3-D posture data," *Ieee transactions on human-machine systems*, vol. 45, no. 5, pp. 586–597, Oct. 2015.

[14] A. De Paola, A. Farruggia, S. Gaglio, G. Lo Re, and M. Ortolani, "Exploiting the human factor in a wsn-based system for ambient intelligence," in *International conference on complex, intelligent and software intensive systems (CISIS '09)*., Mar. 2009, pp. 748–753.

[15] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. New York, NY, USA: Cambridge University Press, 2011.

[16] G. Carlsson, "Topology and data," *Bulletin of the american mathematical society*, vol. 46, no. 2, pp. 255–308, 2009.

[17] N. Ding, L. Melloni, H. Zhang, X. Tian, and D. Poeppel, "Cortical tracking of hierarchical linguistic structures in connected speech," *Nature neuroscience*, vol. 19, no. 1, pp. 158–164, Jan. 2016.

[18] K. S. Fu, *Syntactic methods in pattern recognition*, ser. Mathematics in science and engineering. New York: Academic, 1974, vol. 112.

[19] N. Chomsky, *Syntactic structures*. Walter de Gruyter, 2002.

[20] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages, and computation*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[21] W. Levelt, *An introduction to the theory of formal languages and automata*. John Benjamins Pub., 2008.

[22] N. Chomsky, "Three models for the description of language," *IRE transactions on information theory*, vol. 2, no. 3, pp. 113–124, 1956.

[23] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203–226, 1982.

[24] E. M. Gold, "Language identification in the limit," *Information and control*, vol. 10, no. 5, pp. 447–474, 1967.

[25] M. L. Minsky, *Computation: Finite and infinite machines*. Prentice-Hall, Inc., 1967.

[26] P. Dupont, L. Miclet, and E. Vidal, "What is the search space of the regular inference?" In *In proceedings of the second international colloquium on grammatical inference (ICGI'94*, Springer Verlag, 1994, pp. 25–37.

[27] P. Cottone, S. Gaglio, G. Lo Re, and M. Ortolani, "A machine learning approach for user localization exploiting connectivity data," *Engineering applications of artificial intelligence*, vol. 50, pp. 125–134, 2016.

[28] Z. Balkić, D. Šoštarić, and G. 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*, ser. KES-AMSTA'12, Dubrovnik, Croatia: Springer-Verlag, 2012, pp. 290–298.

[29] D. C. Renso, D. S. Spaccapietra, and D. E. Zimnyi, *Mobility data: Modeling, management, and understanding*. New York, NY, USA: Cambridge University Press, 2013.

[30] Y. Zheng and X. Zhou, *Computing with spatial trajectories*. Springer Publishing Company, Incorporated, 2011.

[31] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of the 17th international conference on world wide web*, ser. WWW '08, Beijing, China: ACM, 2008, pp. 247–256.

[32] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[33] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[34] Y. Chon, H. Shin, E. Talipov, and H. Cha, "Evaluating mobility models for temporal prediction with high-granularity mobility data," in *Proceedings of the 2012 IEEE international conference on pervasive computing and communications (PerCom)*, IEEE, 2012, pp. 206–212.

[35] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles," *ACM Transactions on the Web*, vol. 8, no. 4, 21:1–21:25, Nov. 2014.