AUTOMATISE: Multiple Aspect Trajectory Data Mining Tool Library

Tarlis Tortelli Portela
Federal University of Santa Catarina & University of Pisa & ISTI-CNR & IFPR
tarlis.tortelliportela@isti.cnr.it

Vania Bogorny
Federal University of Santa Catarina
vania.bogorny@ufsc.br

Anna Bernasconi
University of Pisa
anna.bernasconi@unipi.it

Chiara Renso
ISTI-CNR
chiara.renso@isti.cnr.it

***Abstract*—With the rapid increasing availability of information and popularization of mobility devices, trajectories have become more complex in their form. Trajectory data is now high dimensional, and often associated with heterogeneous sources of semantic data, that are called Multiple Aspect Trajectories. The high dimensionality and heterogeneity of these data makes classification a very challenging task both in term of accuracy and in terms of efficiency. The present demo offers a tool, called AUTOMATISE, to support the user in the classification task of multiple aspect trajectories, specifically for extracting and visualizing the *movelets*, the parts of the trajectory that better discriminate a class. The AUTOMATISE integrates into a unique platform the fragmented approaches available in the literature for multiple aspects trajectories and, in general, for multidimensional sequence classification into a unique web-based and python library system. We illustrate the architecture and the use of the tool for offering both movelets visualization and a complete configuration of classification experimental settings.**

***Index Terms*—data mining, python, trajectory classification, trajectory analysis, movelets**

I. INTRODUCTION AND MOTIVATIONS

Mobility data can be represented as a series of spatial information sorted by chronological order recording the moving object position, called *moving object trajectories*. Mobility data representation and analysis have many applications in real life, such as studying the movement of people, vehicles, ships, hurricanes and animals. A typical analysis task is the classification of trajectories like examples the inference of the transportation mode (e.g. car, bus, bike) [2], the strength level of a hurricane [5], a vessel type (e.g. cargo, fish, tourism, etc) [5], or the user of a trajectory [4].

With the rapid increasing availability of information and popularization of mobility devices, trajectories are often associated with heterogeneous sources of semantic data, leading to the concept of Multiple Aspect Trajectories [7], [8] where different semantic dimensions, called *aspects* can be associated to trajectories parts. The heterogeneous dimensions can be of many types, for instance, the spatial dimension type is composed by the latitude and the longitude, that represent a real position in space. The spatial dimension can be associated to time, and can generate numerical features as speed, acceleration, traveled distance, etc. The spatial position also can be associated to a *POI name* or *category* (e.g Hotel, School, Restaurant), and the *price* and *rating* of the place. The trajectories can be associated with the weather condition, user mood, day of the week, and so on. Any data that can be represented as multiple aspect trajectories is compatible with our framework, such as the multivariate time series, therefore from now on we will refer to multiple aspects trajectories also referring to multidimensional sequential data. The classification of such multidimensional objects is a very challenging task both in term of accuracy and efficiency. Movelets has been recently proposed [4] as a possible solution to this challenge. Movelets are subtrajectories of multiple aspects trajectories (thus including a subset of points and/or semantic dimensions) that can represent a class behavior and, therefore, can better discriminate classes in the classification task. Movelets can improve the classification task in efficiency while preserving good accuracy since classifiers can be run on the movelets only disregarding the whole dataset. Movelets can be several in number and complex to understand due to the high dimensionality, therefore a proper tool for visualizing and analyse movelets is needed to give insights about the behaviour of the classes for data analysts.

Frameworks capable of trajectory data visualization as SCIKIT-MOBILITY[1] and GEOPANDAS[2] provide managing and analysis tools based on the raw spatio-temporal data. However, the multiple aspect trajectory analysis, due to intrinsic complexity of the data, need more specific tools that could offer visualization associated with patterns discovered from multidimensional data. Moreover, another fundamental issue is how to make understandable the patterns extracted from such high dimensional data.

Movelet-based trajectory data mining faces the following major challenges:

1) Visualizing the high dimensional data;
2) Visualizing the movelets associated to the trajectory data;

[1]https://github.com/scikit-mobility/scikit-mobility
[2]https://doi.org/10.5281/zenodo.705645

3) Providing the user with a unique platform for accessing the different tools available for movelet extraction and trajectory classification.

This demo paper faces these challenges by introducing the framework AUTOMATISE, an interactive web-system and python library that provides a friendly interface to run multiple aspect trajectory classification and analysis. More precisely, the system provides tools to: (1) configure the experimental environment, (2) select movelets extraction methods, (3) run different classifiers, (4) summarize and visualize the results. As an example, AUTOMATISE can be used by a data analyst for wildlife monitoring, since he/she can visualize and compare movelets extracted from animal trajectories to classify species moving patterns.

It is worth pointing out that the present tool can be useful not only for multiple aspects trajectories but also in a more general setting of Multivariate Time Series data mining.

As a contribution to the research community, we make available AUTOMATISE[3] as an online web application and as a Python library to provide developers and data analysts with a comprehensive platform for classification of multidimensional data.

II. SYSTEM ARCHITECTURE

The AUTOMATISE framework is available as a library in Python that can be easily imported in scripts or notebook coding, and as an interactive web-platform that integrates public datasets, classification methods, and experimental results. The prototype web-platform is currently working with public datasets of multiple aspect trajectories, spatio-temporal trajectories, and time series datasets[4].

The architecture of AUTOMATISE is shown in Figure 1, and it is composed of five main modules: Data Preprocessing, Scripting, Analysis, Results and Visualization Tools described below.

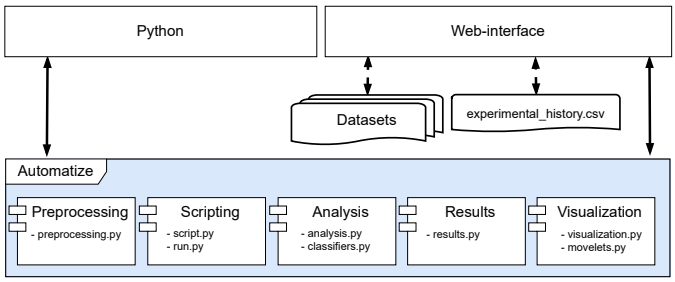

Fig. 1. The architecture of AUTOMATISE platform.

Data Preprocessing: module capable of reading and writing trajectory datasets in three formats: (i) sequential data in comma separated values, (ii) sequential data in cache format zip files, and (iii) multivariate time series files format. It provides methods for hold-out splitting and joining the data, k-fold data split, format conversions, and dataset statistics;

Scripting: the scripting module provide functions to generate command line scripts for running methods and classifiers. Available methods to generate scripts include methods developed in: [1], [3], [4], [6], [8]–[12], but it can be easily extended with other methods;

Analysis: provides python implementation of classifiers for multiple aspect trajectories and movelet-based methods: MARC [6], POI-F [10], Multilayer Perceptron (MLP), Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM) (implemented in [4]);

Results: the module provides functions to read and compile statistics from the experimental resulting files, such as running time and accuracy. The web interface will read a result file precomputed in [8]. Again, this can be easily extended to other results files;

Movelets Visualization: library of tools that provides visualization schemes for movelets as Sankey diagrams[5], Markov chain[6], and a tree visualization.

III. USERS, INTERACTIONS AND DEMONSTRATION SCENARIO

AUTOMATISE has a friendly web-interface that allows the user to interact with most of the functions from the Python library. Thus, the user can perform the following activities: a) trajectory and movelet visualization, b) public dataset review and analysis, c) results exploration, d) experimental environment preparation, and e) related publications review. It is important to highlight that the web-interface scalability is limited to the browser and web server resources. It is a limit that the user has to manage as large datasets will be very hard to load and visualize.

We describe each interface below. However, we emphasize that there is no specific order in the use of these interfaces and they can be employed as the user needs. For example, if the movelets have already been extracted, he/she can open the movelets visualization interface, while if the user needs to configure an experimental environment, he/she can open the Experimental Environment interface.

a) Trajectory and Movelet Visualization: the main features provided by the AUTOMATISE platform are the visualization tools for trajectory data and movelets. The user can drag-and-drop trajectory datasets and movelet files to the Movelet Visualization tool. For instance, the Movelet Sankey diagram in Figure 2 shows movelet sequences, where the first column are the classes and the subsequent columns refer to movelet sequences together. More precisely, the diagram shows the movelet point values in each column and how many movelets contain that value. Thus, following the paths from the first column, the user can visualize the interaction among the different classes.

The AUTOMATISE platform provides the following four types of trajectory and movelet visualizations:

Statistics: display statistics of the trajectories (number of trajectories, attributes, trajectories by class, trajectory sizes,

[3]Platform and source codes: https://github.com/ttportela/automatise

[4]From https://archive.ics.uci.edu/ and http://timeseriesclassification.com/

[5]https://en.wikipedia.org/wiki/Sankey_diagram

[6]https://setosa.io/ev/markov-chains/

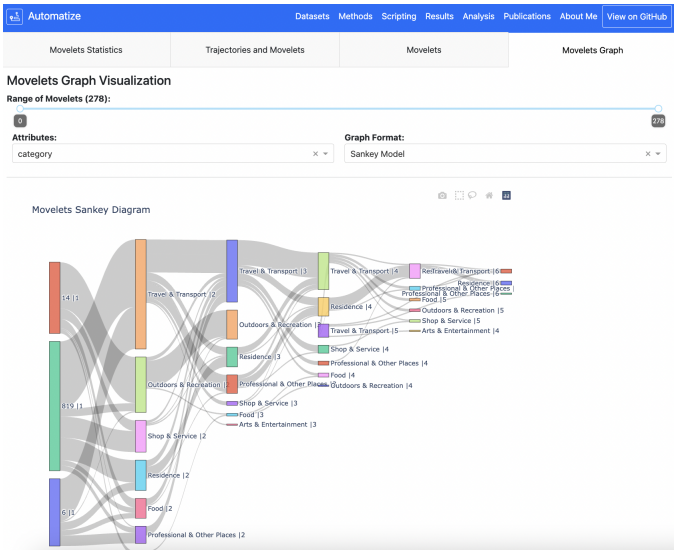

Fig. 2. Movelets visualization interface as Sankey diagram.

and descriptive statistics by each dimension). When importing movelet files, it displays descriptive statistics about the movelets by class, such as the quality, sizes, and used features;

Trajectories and Movelets: displays the trajectory data in each dimension, ordered by the sequence of points. The trajectory points that are present in movelets (for each dimension) are highlighted. The user can interact by selecting the range of trajectories and the dimensions to display. It is important to highlight that the interface allows the user to visualize trajectories and its movelets together;

Movelets: this interface displays the movelets of one selected trajectory (Figure 3). The trajectory points are here displayed aligned with its movelets. This visualization shows the position in the trajectory where the movelets were extracted, or they best fit. The user can look for insights from the movelet patterns that could characterize the class behaviour;

Movelets Graph: provides three forms of graphical visualization of the movelets, where the user can select the movelet range, the dimension to display, and the visualization type: *The Movelet Sankey Diagram* (Figure 2) is a visualization tool used to show the movelet sequences flowing from one set of values to another. It depicts the movelet sequences, starting by the class label, in an interactive way to show how the movelets can inter-connect;

The Movelet Markov Chain shows how the movelet patterns flow from one point to another as a directed graph network. The user can visualize the points, and the sequences that connect these points inside movelets. For example, in Figure 5 we selected the attribute *Category of the Point of Interests* and verified that in the movelets we have a frequent (22) sequence of movements from and to *Outdoor & Recreation*, while other less frequent sequences comes from *Shops & Services* to *Outdoor & Recreation* and from *Arts & Entertainment* to *Outdoor & Recreation*.

The Movelet Tree provides a tree view of the movelets ordered by the higher quality value computed as F-Score [4], and aggregated based on overlapping elements. In other words, it displays the movelets in a top-bottom order of the most relevant subtrajectories to the less significant. It is a simple way of organizing the movelets that can help the user to understand a class behaviour.

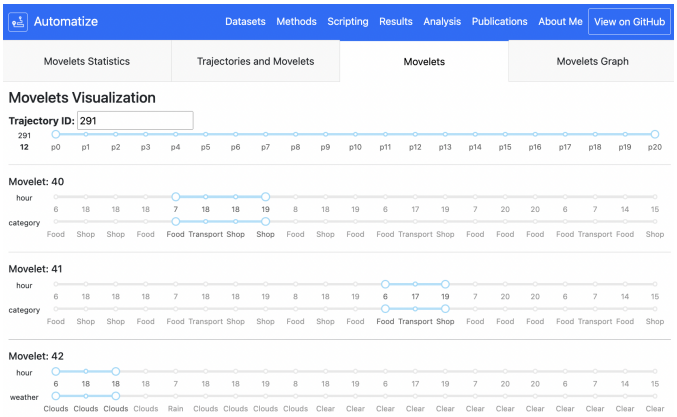

Fig. 3. Movelets visualization screen for selected trajectory.

b) Public Datasets Review and Analysis: the user can browse information about the dataset, related publications, methods and results. The system will dynamically read the available dataset files, classify them by type and associate each one with its descriptions. It dynamically shows published works that uses the dataset, the best method in accuracy and list files to download.

c) Results Exploration: The user can display a predefined results file or can load a new results file in comma separated values. The user can filter results and build rankings of methods by selecting datasets, methods, and classifiers. The rankings are build by accuracy, running time of the method execution and classification running times. The displayed results can be exported to a file.

d) Experimental Environment Preparation: the AUTOMATISE web-interface provides a section to generate an environment for running the movelet extraction methods and the classifiers. The user can configure many parameters to generate the scripts for running the methods. First, it is possible to set a root folder where the environment will be installed. Then, the user can configure the path to datasets, the datasets that will be used, the methods and a timeout for each run (i.e., a time limit for the method to run). The user also can provide the number of threads, memory limit, python specific command, and method specific parameters. Moreover, the user can opt to employ k-fold cross validations. The system will generate all the folder structure, optionally including the methods executable, and the script files to download in one compressed file. (See Figure 4).

e) Related Publications Review: the publications section in web-interface displays aggregated information about the papers for each available classification method. The information in each paper reference is enriched with the datasets used,

the method user guide, examples of use, source codes when available, executable files and citation reference.

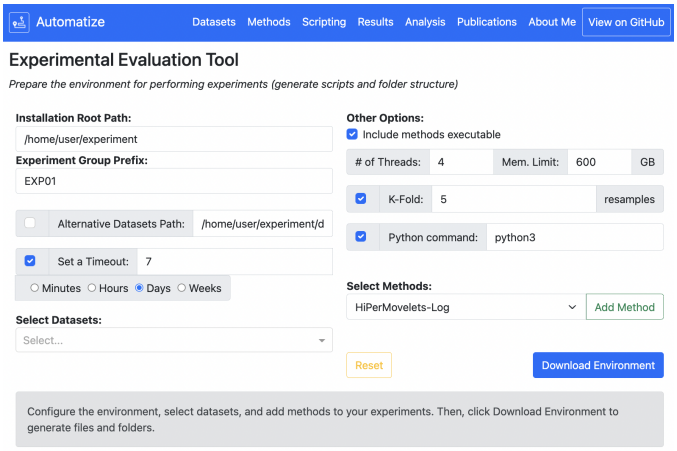

Fig. 4. AUTOMATISE experimental environment preparation.

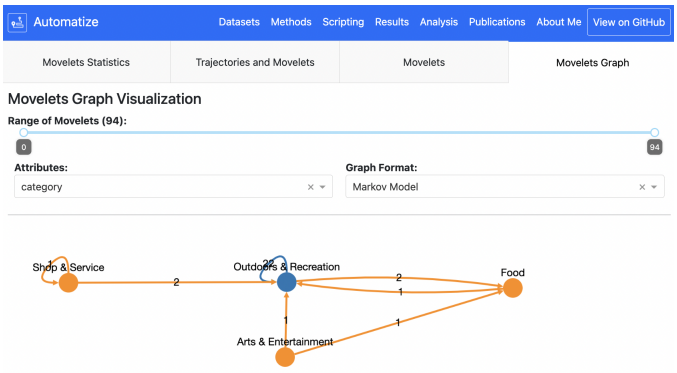

Fig. 5. Movelets visualization as Markov chain.

IV. CONCLUSIONS AND FUTURE WORK

AUTOMATISE is a platform providing both a web interface and a python library that provides tools to perform trajectory and multidimensional data classification task. This demo paper has introduced some functionalities of the python library and the interactions with platform's web version. AUTOMATISE is tailored for multiple aspects trajectories but also to more general multidimensional sequential datasets and offers several options to visualize trajectories and movelets, by hiding many details of the python coding. This speeds up the task of creating interactive visualizations that data analysts need to compare, and offers unique views of the movelets that can give insights to the data analyst.

Multiple aspects trajectories is a model general enough to represent other domains, such as the multivariate time series, event logs or genetic sequences. Thus, AUTOMATISE is an open source platform that was designed to be easily extended for other types of data and feature visualizations. We are extending this prototype to incorporate SCIKIT-MOBILITY, GEOPANDAS, and Time Series visualization tools.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and through the research project Big Data Analytics: Lançando Luz dos Genes ao Cosmos (CAPES/PRINT process number 88887.310782/2018-00). This work was also supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo a Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) - Project Match - co-financing of H2020 Projects - Grant 2018TR 1266 and by the MASTER project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie-Slodowska Curie grant agreement N.777695.

REFERENCES

[1] Somayeh Dodge, Robert Weibel, and Ehsan Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419–434, 2009.

[2] Mohammad Etemad, Amílcar Soares Júnior, and Stan Matwin. Predicting transportation modes of GPS trajectories using feature engineering and noise removal. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1082 LNAI(ii):259–264, 2018.

[3] Carlos Andres Ferrero, Luis Otavio Alvares, Willian Zalewski, and Vania Bogorny. Movelets: Exploring relevant subtrajectories for robust trajectory classification. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 849–856, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Carlos Andres Ferrero, Lucas May Petry, Luis Otávio Alvares, Camila Leite da Silva, Willian Zalewski, and Vania Bogorny. Mastermovelets: discovering heterogeneous movelets for multiple aspect trajectory classification. *Data Min. Knowl. Discov.*, 34(3):652–680, 2020.

[5] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. VLDB Endow.*, 1(1):1081–1094, 2008.

[6] Lucas May Petry, Camila Leite da Silva, Andrea Esuli, Chiara Renso, and Vania Bogorny. MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. *International Journal of Geographical Information Science*, 2020.

[7] Ronaldo dos Santos Mello, Vania Bogorny, Luis Otavio Alvares, Luiz Henrique Zambom Santana, Carlos Andres Ferrero, Angelo Augusto Frozza, Geomar Andre Schreiner, and Chiara Renso. MASTER: A multiple aspect view on trajectories. *Transactions in GIS*, 2019.

[8] Tarlis Tortelli Portela, Jonata Tyska Carvalho, and Vania Bogorny. Hipermovelets: high-performance movelet extraction for trajectory classification. *International Journal of Geographical Information Science*, 0(0):1–25, 2022.

[9] Tarlis Tortelli Portela, Camila Leite da Silva, Jonata Tyska Carvalho, and Vania Bogorny. Fast movelet extraction and dimensionality reduction for robust multiple aspect trajectory classification. In André Britto and Karina Valdivia Delgado, editors, *Intelligent Systems*, pages 468–483, Cham, 2021. Springer International Publishing.

[10] Francisco Vicenzi, Lucas May Petry, Camila Leite Da Silva, Luis Otavio Alvares, and Vania Bogorny. Exploring frequency-based approaches for efficient trajectory classification. *Proceedings of the ACM Symposium on Applied Computing*, pages 624–631, 2020.

[11] Zhibin Xiao, Yang Wang, Kun Fu, and Fan Wu. Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, 6(2), 2017.

[12] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei Ying Ma. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web*, 4(1), 2010.