TourMiner: Effective and Efficient Clustering Big Mobile Social Data for Supporting Advanced Analytics Tools

Gloria Bordogna[1], Alfredo Cuzzocrea[2,3],
Luca Frigerio[1], Giuseppe Psaila[4], Danilo Amendola[5]

[1]CNR-IREA, Milano, Italy
[2]DIA Dept., University of Trieste, Trieste, Italy
[3]ICAR-CNR, Rende, Italy
[4]DIGIP Dept., University of Bergamo, Bergamo, Italy
[5]IRCCS Bonino-Pulejo, Messina, Italy
bordogna.g@irea.cnr.it; alfredo.cuzzocrea@dia.units.it;
frigerio.l@irea.cnr.it; psaila@unibg.it; daniloamendola@gmail.com

Abstract. Nowadays a great deal of attention is devoted to the issue of supporting *big data analytics* over *big mobile social data*. These data are generated by modern emerging social systems like *Twitter*, *Facebook*, *Instagram*, and so forth. *Mining big mobile social data* has been of great interest, as analyzing such data is critical for a wide spectrum of big data applications (e.g., *smart cities*). Among several proposals, *clustering* is a well-known solution for extracting interesting and actionable knowledge from massive amounts of big mobile (geo-located) social data. Inspired by this main thesis, this paper proposes *an effective and efficient similarity-matrix-based algorithm for clustering big mobile social data*, called TourMiner, which is specifically targeted to *clustering trips* extracted from *tweets*, in order to mine *most popular tours*. The main characteristic of TourMiner consists in applying clustering over a well-suited *similarity matrix* computed on top of trips.

1 Introduction

Social networks coupled with the diffusion of *smart mobile devices* equipped with GPS sensors have contributed to generate a big amounts of users-generated contents which are geo-referenced and timestamped. These are recognized in literature as *big mobile social data* (e.g., [1,2,3]). By analizying recent research trends, it follows that, nowadays, a great deal of attention is devoted to the issue of supportingbig mobile social data analytics. These data are generated by modern emerging social systems like *Twitter*, *Facebook*, *Instagram*, and so forth. *Mining big mobile social data* (e.g., [13,14]) has been of great interest, as analyzing such data is critical for a wide spectrum of big data applications (e.g., *smart cities*). The possibility of cross-analyzing geo-references, timestamps, characteristics of content authors, and, last but not least, message contents, poses new challenges to the research community (e.g., [4]). CISCO estimated that data on the Internet will increase at a *Compound Annual Growth Rate* of 25% by the year 2017. Thus, in order to deal with continuously-growing-in-size data sets, it will be necessary to frequently scale-up existing algorithms or to define new paradigms for *big mobile social data analytics* (e.g., [5,6]).

Another aspect to be faced-off is represented by the amenity of *synthesizing* big data collections with the purpose of highlighting *relevant* meanings embedded within them.

Several real-world applications, such as *social network community identification* (e.g., [7]) and their *trajectory summarization* (e.g., [8]), require means to filter-out the communities of users and further to group and represent similar trajectories. This latest task implies scaling-down the collecions of big data by removing noise and redundancies, so that highlighting the main representative *spatio-temporal patterns* (e.g., [9]). *Faceted search* (e.g., [10]) and *clustering techniques* can be used to this purpose since their goal is to identify groups of items within the target data set with common characteristics in a *feature space*, while removing outliers and noise which are considered uninteresting from further analysis.

Inspired by this main thesis, this paper proposes *an efficient implementation of similarity-matrix-based algorithm for clustering big mobile social data*, called TourMiner, which is specifically targeted to *clustering trips* extracted from geo-located *tweets*, in order to mine *most popular tours*. The main characteristic of TourMiner consists in applying clustering (any clustering algorithm) over a well-suited *similarity matrix* (possibly using distinct similarity measures) computed on top of trips.

Looking at specific algorithmic charateristics, TourMiner can be considered as a *hybrid algorithm* as it sequentially applies a "scaling-up" and subsequently a "scaling-down" phase to mine tours. In particular, the first phase applies a scale-up approach in order to identify the community of tourists visiting a region of interest within a big set of Tweet messages, which are collected during a period of time, so that data increasingly scale up with time, thus efficiently improving our previous contribution [15]. The second phase, which is based on *geo-partitioning* and *geo-clustering*, applies a scale-down approach for identifying and synthetising the main tours followed by most of the tourists within the target region, by exploiting low-cost hardware, *GPU-based computing paradigms* (e.g., [16]) and open-source libraries to manage available memory resources, thus significantly extending and further improving our previous contribution [17] and clearly matching the strict computational requirements of big data management by means of low cost hardware (e.g., [18]).

2 TourMiner: An Effective and Efficient Algorithm for Mining Most-Popular Tours from Big Social Media

In this Section, we provide principles and implementation of the proposed algorithm TourMiner. TourMiner comprises three main phases:

1. *Geo-partitioning*: trips extracted from tweets are generated according to different representations so that they are defined as semantic trajectories over a spatial domain ontology, intended as a representation of spatial entities of interest with their geographic footprints;
2. *Computing the pairwise trip similarity matrix*: trips are processed in order to compute a suitable similarity matrix that is the fundamental data structure of our clustering approach;
3. *Trip clustering (based on the similarity matrix)*: finally, on the basis of the similarity matrix, trips are clustered as to extract actionable knowledge from the target big mobile social media data.

In the following, we describe these phases in details.

2.1 Geo-Partitioning

Firstly, each trip is represented by a *semantic trajectory*, i.e., a sequence of IDs $\langle a_1, a_2, \ldots, a_n \rangle$ that identify interesting locations on the spatial domain, such as Points Of Interest (POIs) like restaurants, hotels, museums, etc, or area districts such as city areas, municipalities, etc. An example of semantic trajectory defined by sequence of IDs could be:

<Malpensa Airport, Milano Central Station, Bergamo Railway Station, Piazza Vecchia>

By this representation we may argue that the traveller landed in Malpensa airport, then went to Milano Central Station and from there he travelled by train to Bergamo where he visited the medioeval Piazza Vecchia.

This same trajectory could be represented in a coarsest way bearing more uncertainty on the visited locations, for example by a sequence of municipalities, namely:

<Malpensa, Milano, Bergamo, Bergamo>

In this case, we can just guess the visited cities.

In this phase, one can choose to represent a trip based on the following two kinds of IDs:

a) Categorical IDs: trips are represented in terms of a sequence of *objects of interest* that are implemented as *Point Of Interest* (POI) on the spatial domain or zones such as municipalities, ZIP code areas, NUTS etc. In this case, IDs are categorical and neither a metric, nor an order is defined between them.

b) Numerical IDs: trips are represented in terms of a sequence of *natural numbers* $a_i \in N$ among which a linear order is defined so that $a_i < a_j$ if $i < j$. These IDs identify cells in a *regular tessellation* of the spatial domain, and can be generated by *space filling functions* [12] such as the *Z-Order* and the *Peano* curve. They are named as *GeoHash codes* and a metric over them is defined by a function of kind: $|.| \; N \rightarrow N$ that satisfies the following properties: (*i*) $|a_i - a_i| = 0$; (*ii*) $|a_i - a_j| = |a_j - a_i|$; (*iii*) $|a_i - a_j| + |a_j - a_k| \geq |a_i - a_k|$.

The advantage of using GeoHash codes is represented by the opportunity of using *similarity measures* based on a *metric distance*, thus achieving more accuracy. In fact, when using categorical IDs, the matching operation between two IDs is crisp, they either match or not, and, in this context, it is meaningless to evaluate *fuzzy matching*, since two IDs, e.g. ZIP codes, may differ only by one digit and be distant in space. On the other hand, when using GeoHash codes, one obtains trajectoies whose semantic is both implicit and uncertain with respect to the objects of interest: to explicit the semantics, one should determine which POIs are within each cell identified by the GeoHash codes.

2.2 Computing the Pairwise Trip Similarity Matrix

Computing the pairwise trip similarity matrix depends on the types of IDs used to represent trips (see Section 2.1), i.e. via categorical IDs or numerical IDs. In the following, we address both cases. Before computing such similarity matrix, we first

eliminate *redundancies*, i.e. *consecutive duplications* of the same ID in the trip representation. Therefore, a trip of kind:

$$A = \langle ID_1, ID_2, ID_3, ID_3, ID_3, ID_4, ID_3 \rangle$$

becomes of kind:

$$A = \langle ID_1, ID_2, ID_3, ID_4, ID_3 \rangle$$

after redundancy removal.

Furthermore, *pointwise trips* are deleted. For instance, a trip of kind:

$$B = \langle ID_1, ID_1, ID_1 \rangle$$

becomes of kind:

$$B = \langle ID_1 \rangle$$

after redundancy removal and then, being pointwise, it is deleted.

The motivation of redundancy removal relies in the idea of not to favor the subsequent grouping of trips into a tour just because many tweets have been sent from the same locality. This usually happens in the airport area, and free WIFI hotspots .

We next describe how the pairwise trip similarity matrix is defined for both cases (i.e., categorical IDs and numerical IDs).

Case 1: Categorical IDs. Let $A = \langle a_1, a_2, \ldots, a_N \rangle$ and $B = \langle b_1, b_2, \ldots, b_M \rangle$ denote a pair of trips of different length, such that $N \neq M$, being a_i and b_j strings that identify categories (i.e., ZIP or NUTS codes). Let $A \cap B$ denote the intersection trip between A and B. We define the cardinality of $A \cap B$, denoted by $|A \cap B|$, as follows:

$$|A \cap B| = \sum_{i=1}^{\min(N,M)} \max_{j=1,\ldots \max(N,M) \mid bj \neq ak \; \forall \; k<i} (a_i = b_j) \quad (1)$$

such that:

$$a_i \in A \; with \; |A| = \min(N, M) \quad (2)$$

and:

$$b_j \in B \; with \; |B| = \max(N, M) \quad (3)$$

Formula (1) is computed as follows. We first scan every element of the *shortest* trip A and match it with every element of the *longest* trip B by increasing the counter if there is a new element in the longest trip (not previously matching any other element of B) that matches the current element of A. This way, we constrain an element of B to be selected only once by one single element of A. This measure is *symmetric*. This means that by scanning the longest trip and matching it with every element of the shortest one of the two respective trips we would obtain the *same* result due to the fact that we cancel, at each match, the elements already selected.

Furthermore, the cardinality of intersection trip satisfies the following properties:

$$\begin{gathered} |A \cap B| = |B \cap A| \\ |A \cap A| = |A| \end{gathered} \quad (4)$$

Finally, in our framework, we exploit the following similarity measures between trips (case of trips modeled in terms of categorical IDs):

$$
\begin{aligned}
&Fuzzy\_Inclusion(A,B) = |A \subseteq B| = \frac{|\mathrm{A} \cap \mathrm{B}|}{\min(N,M)} \\
&Jaccard\_Sim(A,B) = \frac{|\mathrm{A} \cap \mathrm{B}|}{N + M - |\mathrm{A} \cap \mathrm{B}|} \\
&Dice\_Sim(A,B) = 2\frac{|\mathrm{A} \cap \mathrm{B}|}{N + M} \\
&Cosine\_Sim(A,B) = \frac{|\mathrm{A} \cap \mathrm{B}|}{\sqrt{N} * \sqrt{M}} \\
&Fuzzy\_Sim(A,B) = \frac{1}{2}\left(\frac{|\mathrm{A} \cap \mathrm{B}|}{N} + \frac{|\mathrm{A} \cap \mathrm{B}|}{M}\right)
\end{aligned}
\tag{5}
$$

These similarity measures are used to enrich the knowledge discovery phase from big social media implemented in our framework.

Case 2: Numerical IDs. Let $A = \langle a_1, a_2, \ldots, a_N \rangle$ and $B = \langle b_1, b_2, \ldots, b_M \rangle$ denote a pair of trips of different length, such that $N \neq M$, being a_i and b_j numbers that identify cells (i.e., GeoHash codes, which we assume to be implemented in terms of Z-Order codes – see Section 2.1). Let *MaxZcode* be the maximum Zcode among those generated, which is defined as follows:

$$MaxZcode = \max(a_N, b_M) \tag{6}$$

Also, let $(A - B)$ denote the difference-trip between A and B, and $|A - B|$ its cardinality, which is defined as follows:

$$|A - B| = \sum_{i=1}^{N} \min_{\substack{j=1,\ldots,M \mid \\ b_j \neq \mathrm{argmin}(|a_k - b_j|) \forall k < i}} (|a_i - b_j|) \tag{7}$$

Furthermore, the cardinality of difference-trip satisfies the following properties:

$$
\begin{aligned}
|A - B| &= |B - A| \\
|A - A| &= 0
\end{aligned}
\tag{8}
$$

Finally, in our framework, similarly to the case of trips modeled as categorical IDs, we exploit the following similarity measures between trips (case of trips modeled in terms of numerical IDs):

$$
\begin{aligned}
&Fuzzy\_Inclusion\_Hash(A,B) \\
&\qquad = \frac{MaxZcode - \frac{|A - B|}{\min(N,M)}}{MaxZcode} \\
&Jaccard\_Sim\_Hash(A,B) \\
&\qquad = \frac{MaxZcode - \frac{|A - B|}{N + M - |A - B|}}{MaxZcode} \\
&Dice\_Sim\_Hash(A,B) = \frac{MaxZcode - 2\frac{|A - B|}{N + M}}{MaxZcode}
\end{aligned}
\tag{9}
$$

$$Cosine\_Sim\_Hash(A,B) = \frac{MaxZcode - \frac{|A-B|}{\sqrt{N} * \sqrt{M}}}{MaxZcode}$$

$$Fuzzy\_Sim\_Hash(A,B) = \frac{MaxZcode - |A-B|}{MaxZcode}$$

These similarity measures, just like to case of trips modeled as categorical IDs, are used to enrich the knowledge discovery phase from big social media implemented in our framework.

On the basis of the definitions above, the pairwise trip similarity matrix *SIM* is introduced as a $D \times D$ matrix such that, for each pair of trips t_i and t_j, the entry $SIM(t_i,t_j)$ satisfies the following properties:

$$i.\ SIM(t_i, t_i) = 1 \qquad (10)$$
$$ii.\ SIM(t_i, t_j) = SIM(t_j, t_i)$$

Based on (*ii*), the similarity matrix *SIM* is a *triangular matrix* hence we need to compute only its values below the main diagonal, i.e. $\forall\ i = 1, \ldots, D$. We compute $SIM(t_i,t_j)$ with $j = 1,\ldots, i\text{-}1$ only.

Moreover, since the computation of the similarity between any two trips is independent from the other trips, in our implementation this process has been parallelized by exploiting GPU-based computing paradigms.

2.3 Similarity-Matrix-based Trip Clustering

Here we describe the core strategy of our proposed TourMiner algorithm, i.e. clustering of trips based on the similarity matrix *SIM* (see Section 2.2).

In this phase, clustering is applied to group trips into clusters based on their similarity, and then the biggest clusters are selected as popular tours. In principle, any clustering algorithm based on a similarity matrix could be applied. In our implementation, we tested both the density based *DBSCAN clustering* algorithm, generating a flat partition [24], and the *hierarchical agglomerative complete link clustering* whose main idea consists in generating, at each hierarchical cycle, a new cluster by focusing on the two clusters, said C_i and C_j , with greatest similarity in *SIM*. The *complete link clustering* main steps are the following:

Step 1. Identification of the two clusters to fuse C_i and C_j as follows:

$$\{i,j\} = \text{argmax}(SIM(i,j))\ \ i,j = 1,\ldots,D \qquad (11)$$

Step 2. Computation of the similarity of the actual clusters with respect to the candidate cluster in terms of the *minimum* among the similarities with the elements of the candidate cluster (see formula in step 4.3 in of Algorithm 1). This guarantees that, when a new element or cluster is added to a cluster to form a higher-level cluster, all its (clustered) elements share a *minimum* similarity degree. This allows generating compact clusters.*Step 3.* Selection of the popular tours: they are the first k biggest clusters in partition with minimum similarity ε, being k and ε tunable input parameters. ε identifies the level of the cluster hierarchy, i.e., the cluster partition. k is used to select the clusters with greatest number of trips from this partition.

3 Experimental Assessment and Analysis

In order to prove the effectiveness and the efficiency of our proposed algorithm TourMiner, we designed and developed an experimental campaign, whose results are reported in this Section.

In particular, experiments were aimed at evaluating the proposed clustering approach in terms of both time and memory needed to process the distinct phases (see Section 2). We designed and evaluated optimized solutions for all these phases. We discuss these solutions in the following.

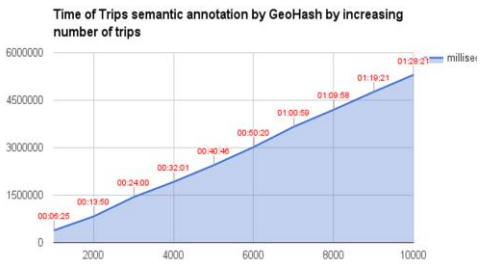

Fig. 1. Tip Creation Time

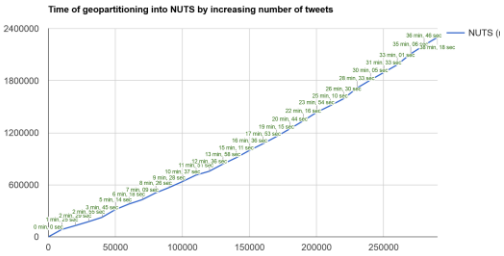

Fig. 2. Geo-Partitioning with NUTS

We developed ad-hoc metrics over such phases, and we studied the evolution of such metrics by increasing the number of tweets and the number of trips. Both parameters well-test the scalability of our proposed algorithm. In the following, we provide our experimental results.

First, in order to have a better understanding on the realibility of our experimental campaign, Fig. 1 shows the time needed to create annotated trips, i.e., semantic trajectories, in terms of GeoHash codes depending on the number of tweets. This, indeed, allows us to better estimate the size of data we deal with, as size is one of the main characteristics of big data to be considered when designing algorithms that run on big data sets.

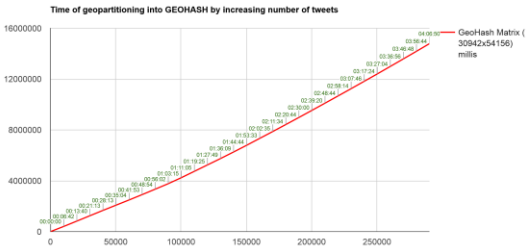

Fig. 3. Geo-Partitioning with GeoHash

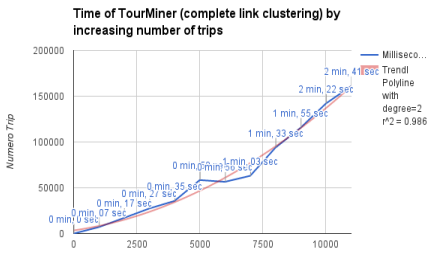

Fig. 4. Temporal Perfomance of the Whole Tour Mining Process with GeoHash and Complete-link Clustering

Fig. 2 and Fig 3 focus, instead, on the geo-partitioning phase of algorithm TourMiner. Here, we report the time needed for computing the geo-partition in the two different cases of NUTS (Fig. 2) and GeoHash (Fig. 3), still with respect to the number of tweets without the parallel optimization.

Fig. 4 reports the time needed for the *whole* tour mining process by using GeoHash codes as semantic annotation of trips and the Complete-link clustering algorithm to identify popular tours. Here, it should be noticed that our introduced optimization support for the geo-partitioning phase (see Section 2) allows us to obtain a polynomial trend with degree 2 instead of 3. This further confirms to us the realibility of our clustering methodology over big mobile social data.

By analyzing the experimental results of our campaign, it clearly follows the efficiency of our proposed algorithm TourMiner for clustering big mobile social data.

4 Conclusions and Future Work

In this paper, we have introduced and experimentally assessed TourMiner, an effective and efficient algorithm for spatially mining big mobile social data about to semantic trajectories. TourMiner implements a clustering procedure that founds on (*i*) suitable semantic representations of trips extracted from Twitter data, alternatively in terms of categorical IDs or numerical IDs, and (*ii*) a well-suited similarity matrix computed on top of such trips. We provided motivations, definitions and analysis of TourMiner, along with its experimental assessment and analysis over Twitter data according to several (experimental) parameters. Results clearly comfirm the benefits coming from our proposal.

Future work is mainly oriented towards enriching our proposed framework with innovative characteristics like *uncertain big data management* (e.g., [19]) and *big data approximation paradigms* (e.g., [27]).

References

[1] S. M. Allen, M. J. Chorley, G. Colombo, E. Jaho, M. Karaliopoulos, I. Stavrakakis, R. M. Whitaker, "Exploiting User Interest Similarity and Social Links for Micro-Blog Forwarding in Mobile Opportunistic Networks", *Pervasive and Mobile Computing 11*, pp. 106–131, 2014

[2] O. Khalid, M. U. S. Khan, S. U. Khan, A. Y. Zomaya, "Omnisuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks", *IEEE Transactions on Services Computing 7(3)*, pp. 401–414, 2014

[3] Y. Yu, X. Huang, X. Zhu, G. Wang, "Camel: A journey group t-pattern mining system based on instagram trajectory data", in: *Proceedings of IEEE DASFAA*, pp. 527–530, 2014

[4] M. Balduini, A. Bozzon, E. D. Valle, Y. Huang, G. Houben, "Recommending Venues using Continuous Predictive Social Media Analytics", *IEEE Internet Computing 18(5)*, pp. 28–35, 2014.

[5] Alfredo Cuzzocrea, Domenico Saccà, Jeffrey D. Ullman, "Big data: a research agenda", in: *Proceedings of IDEAS 2013*, pp. 198–203, 2013

[6] A. Cuzzocrea, "Analytics over Big Data: Exploring the Convergence of Data Warehousing, OLAP and Data-Intensive Cloud Infrastructures", in: *Proceedings of COMPSAC 2013*, pp. 481–483, 2013

[7] M. Takaffoli, R. Rabbany, O. R. Zaïane, "Incremental Local Community Identification in Dynamic Social Networks", in: *Proceedings of ASONAM*, pp. 90–94, 2013

[8] H. Su, K. Zheng, K. Zeng, J. Huang, S.W. Sadiq, N.J. Yuan, X. Zhou, "Making Sense of Trajectory Data: A Partition-and-Summarization approach", in: *Proceedings of IEEE ICDE*, pp. 963–974, 2015

[9] H. Jiang, L. Huang, J.J. Zhang, Y. Zhang, D.W. Gao, "Spatialtemporal Characterization of Synchrophasor Measurement Systems – A Big Data Approach for Smart Grid System Situational Awareness", in: *Proceedings of ACSSC*, pp. 750–754, 2014

[10] H. Zhuge, Y. Wilks, "Faceted Search, Social Networking and Interactive Semantics", *World Wide Web 17(4)*, pp. 589–593, 2014

[11] R.C. Daley, J.B. Dennis, "Virtual Memory, Processes, and Sharing in MULTICS", *Communications of the ACM 11(5)*, pp. 306–312, 1968

[12] M.F. Mokbel, W.G. Aref, I. Kamel, "Analysis of Multi-Dimensional Space-Filling Curves", *GeoInformatica 7(3)*, pp. 179–209, 2003

[13] S. Jin, W. Lin, H. Yin, S. Yang, A. Li, B. Deng, "Community Structure Mining in Big Data Social Media Networks with MapReduce", *Cluster Computing 18(3)*, pp. 999–1010, 2015

[14] N.J. Yuan, "Mining Social and Urban Big Data", in: *Proceedings of ACM WWW (Companion Volume)*, p. 1103, 2015

[15] A. Cuzzocrea, G. Psaila, M. Toccu, "An Innovative Framework for Effectively and Efficiently Supporting Big Data Analytics over Geo-Located Mobile Social Media", in: *Proceedings of ACM IDEAS*, 2016

[16] J.A. Cedillo, I. Rivera-Zarate, J.C. Herrera Lozada, M. Olguin-Carbajal, "GPU Programming Paradigm", *Journal of Theoretical and Applied Information Technology*, pp. 133–138, 2009

[17] G. Bordogna, A. Cuzzocrea, L. Frigerio, G. Psaila, "Clustering Geo-Tagged Tweets for Advanced Big Data Analytics", in: *Proceedings of IEEE BigData Congress*, 2016

[18] B. Yu, A. Cuzzocrea, D.H. Jeong, S. Maydebura, "On Managing Very Large Sensor-Network Data Using Bigtable", in: *Proceedings of CCGRID 2012*, pp. 918–922, 2012

[19] A. Cuzzocrea, C.K.S. Leung, R.K. MacKinnon, "Mining Constrained Frequent Itemsets from Distributed Uncertain Data", *Future Generation Computer Systems 37 (1)*, 117–126, 2014

[20] F. Giannotti, M. Nanni, F. Pinelli, D. Pedreschi, "Trajectory Pattern Mining", in: *Proceedings of ACM SIGKDD*, pp. 330–339, 2007

[21] L.O. Alvares, V. Bogorny, B. Kuijpers, J.A.F. de Macedo, B.Moelans, A. Vaisman, "A Model for Enriching Trajectories wth Semantic Geographical Information", in: *Proceedings of ACM GIS*, 2007.

[22] S. Spaccapietra, C. Parent, M.L. Daminai, J.A.F. de Macedo, F. Porto, C. Vangenot, "A Conceptual View on Trajectories", *IEEE Transactions on Knowledge and Data Engineering*, 2008

[23] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, K. Aberer, "Semitri: A Framework for Semantic Annotation of Hetherogeneous Trajectories", in: *Proceedings of EDBT*, 2011

[24] M. Ester, H.P. Kriegel, J. Sander, X. Xu, "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in: *Proceedings of ACM KDD*, 1996

[25] Z. Yin, L. Cao, J. Han, J. Luo, T.S. Huang, "Diversified Trajectory Pattern Ranking in Geo-Tagged Social Media", in: *Proceedings of SDM*, pp. 980–991, 2011

[26] C. Comito, D. Falcone, D. Talia, "Mining Popular Travel Routes from Social Network Geo-Tagged Data", in: *(E. Damiani et al. eds), "Intelligent Interactive Multimedia Systems and Services", Springer*, pp. 81–95, 2015

[27] A. Cuzzocrea, U. Matrangolo, "Analytical synopses for Approximate Query Answering in OLAP Environments", in: *Proceedings of DEXA*, pp. 359–370, 2004