Detection of Points of Interest in a Smart Campus

Article

Accepted version

A. De Paola, A. Giammanco, G. Lo Re, G. Anastasi

In Proceedings of the 5th International Forum on Research and Technologies for Society and Industry (RTSI 2019)

Publisher: IEEE

Detection of Points of Interest in a Smart Campus

Alessandra De Paola, Andrea Giammanco and Giuseppe Lo Re
University of Palermo
Palermo, Italy
{alessandra.depaola, andrea.giammanco, giuseppe.lore}@unipa.it

Giuseppe Anastasi
University of Pisa
Pisa, Italy
giuseppe.anastasi@unipi.it

***Abstract*—Understanding users' habits is a critical task in order to develop advanced services, such as personalized recommendation and virtual assistance. In this work, we propose a novel approach to detect Points of Interest visited by users of a campus, by using mobility traces collected through users' smartphones. Our method takes advantage of the intentional and recurrent nature of human movements to build up mobility profiles, and combines different machine learning methods to merge sensory information with the past users' behavior. The proposed approach has been validated on a synthetic dataset and the experimental results show its effectiveness.**

***Index Terms*—Smart Campus, PoI Automatic Detection, Human Mobility Profiling**

I. Introduction

As the diffusion of personal devices increases, it is becoming easier to keep track of users' trajectories. In many fields, such as pervasive computing and social sciences, user profiling is a very valuable task, since it enables the creation of models of their activities, seen as sequences of movements. As proved in [1], human trajectories display a high degree of temporal and spatial regularity, and each user is characterized by a significant probability to return to a few highly frequented locations. Therefore, an accurate location identification makes location-aware applications more effective.

Context awareness is the key feature of *Smart Environments* [2], and in particular of a *Smart Campus* [3], [4], which is a digitally augmented campus where pervasive instrumented objects and spaces are made responsive to the state of the environment and its inhabitants. Location-aware computing aims at extracting information from raw trajectory data, in order to supply personalized services. In a Smart Campus, in addition to the ubiquity of users' smartphones, several other IoT sensory devices [5], such as cameras, RFID readers, and bluetooth beacons, collect raw measurements, that can be exploited by an intelligent system in order to reason upon current context and supply advanced services to users. A location-based recommender system can provide information relevant to users' position, e.g. suggesting the nearest free library seat. Moreover, the capability of predicting users location can be exploited to provide recommendations related to the next place a user will visit, e.g., enabling the suggestion of free parking space near the next destination.

The work described here aims at inferring Points of Interest (PoIs) visited by users, using georeferenced position data. The regions where a user stops for a considerable amount of time can be automatically extracted by means of clustering [6], [7]. Nevertheless, mapping a user's position to a set of known PoIs is not a trivial task [8]–[10], due to the intrinsic error in measurements and the presence of areas dense of meaningful places. The user's past behavior can be used to refine the estimation of their location, but the cold-start problem for new users has to be faced. We propose to face these issues, by adopting a combination of unsupervised and supervised machine learning methods, in order to be capable of dealing with uncertain data and of merging sensory data with knowledge about past user's behavior.

The remainder of the paper is organized as follows. Sec. II reviews the literature about automatic methods for detecting PoIs. Sec. III outlines the architecture of the proposed system, by providing a high level description of its components. Sec. IV describes the algorithm proposed to identify visited PoIs. Sec. V describes the experimental evaluation performed to assess the performance of the proposed approach. Finally, Sec. VI states some conclusions and discuss the future work.

II. Related Work

Several methods have been described in the literature for mining significant locations from georeferenced data, and they can be grouped into two classes [6]: *geometry-based* and *fingerprint based* methods. *Geometry-based* methods use clustering techniques on raw position data, e.g., coming from GPS sensors, to produce coordinates or polygonal shapes describing users' significant places. *Fingerprint based* methods exploit a set of *fingerprint waypoints*, which are defined as "signatures" of different places. Through such signatures, a user's personal device is able to detect when it returns to a place already visited, without knowing its geographical location.

Moreover, the approaches described in the literature can be analyzed with respect to their capability of performing indoor or outdoor PoI identification, or with respect to the capability of detecting PoIs in real time. An overview of some relevant approaches is summarized in Table I, which also highlights the clustering method used to identify the users' *Stop Points*.

Authors of [11] group stop points using an enhanced version of OPTICS clustering algorithm, which performs multiple split-and-merge steps until any single cluster represent an unique semantic location. Authors of [12] focus on the time difference between consecutive GPS samples, detecting a stop point whenever a given time interval occurs between them. The key insight is that GPS signals can not be collected inside buildings, thus an event of signal drop corresponds to

TABLE I
OVERVIEW OF POI DETECTION METHODS

Qualitative categories: (a) Geometry-based or Fingerprint-based; (b) Indoor/Outdoor relevant places, or Both; (c) Online or Offline.

Approach	Qualit. categories (a)	(b)	(c)	Clustering Method
[11]	G	B	Off	OPTICS
[12]	G	I	Off	k-means
[13]	G	B	On	k-means / GMM
[14]	G	B	Off	spatio-temporal
[15]	F	I	Off	robust beacon infer.

a prolonged visit in a place. Gathered GPS readings are then clustered through the k-means algorithm to extract meaningful locations. Such an approach could lead to erroneous detection of stop points in areas with discontinuous signals. The approach proposed in [13] exploits the periodical transmission of Wi-Fi beacons performed by access points. Through received beacons, users can detect their location by averaging the access points' locations. In order to deal with possible sensory errors, relevant places are detected as clusters of locations, identified through k-means clustering and Gaussian mixture models. Authors of [15], instead, exploit Wi-Fi beacons to detect entering and leaving to and from a place. Authors of [14] introduce a combination of spatial and temporal constraints in order to detect stop points; the proposed approach iteratively analyzes the spatial regions where the user stops for a a given time period, and detects a stop if a user spends more than 30 minutes within a range of 200 meters.

Despite plenty of work has been done to discover PoIs, there are still open issues to be addressed, mainly in order to combine raw measures with other high level information, so thus to increase the detection accuracy [8]–[10]. Differently from other previous works, our approach aims at detecting PoIs by adopting a probabilistic and dynamic approach, in order to merge sensory data, which can be noisy and inaccurate, with context information related to users' activities and habits.

III. SYSTEM ARCHITECTURE

Our Smart Campus system relies on a multi-tier architecture that allows to extract relevant information from raw data in order to provide context-aware services. Fig. 1 shows an overview of such architecture, with a detailed focus on the role played by the *PoI Detection* subsystem.

The lowest layer of our architecture detects relevant events and monitors physical phenomena through a pervasive sensory infrastructure [16]. The *PoI Detection* subsystem analyzes raw sensory data to detect a set of *Stop Points*, which can be defined as regions where the user remains for a certain amount of time, and merges the identified *Stop Points* with high-level context information in order to detect the actual visited PoIs. Such context information summarizes the relationship between users and the activities they perform in different

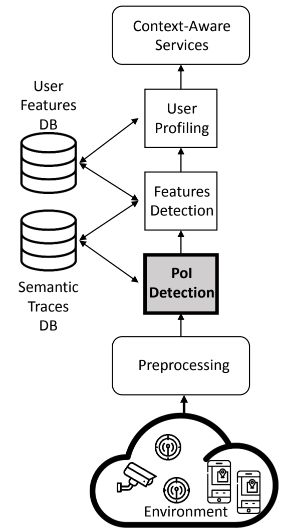

Fig. 1. Architecture of the proposed Smart Campus.

PoIs, in terms of frequency of visit, average stop time, and sequence of visited PoIs. The detected PoIs are sent to the *Features Detection* module in order to extract a set of n-dimensional points which summarize the mobility behavior of each user, such as his leaning to be sedentary or exploratory during the workday. The *User Profiling* module then aims at outlining different classes of users according to previously extracted features. Besides inferred behavioral characteristics, user profiles also include information explicitly provided by the users themselves, such as the their role (e.g., "researcher" or "student"). Finally, knowledge about user profiles and their current and next position and activity can be exploited in order to provide users a set of *Context-Aware Services*.

IV. POI AUTOMATIC DETECTION

The main contribution of this paper is the PoI detection algorithm, which merges sensory data with context information, in order to solve possible ambiguities deriving from noisy or discontinuous readings.

A PoI is defined as a place where the user usually goes and stops for a while. It could be a place of interest for the whole community, such as a shop or a bus stop, or for a single user. Typical PoIs inside a campus include departments, libraries, parks, research laboratories, auditoriums and lecture halls.

In this work, we address the *personalized automatic check-in problem* [8]: given a PoI database and a set of georeferenced user's data, we want to recover the sequence of PoIs the user visited. Such problem can be divided into two sub-problems: the *Stop Points* discovery from data, and the *Checked-in PoI* detection for each stop point. Therefore, it is necessary to formally define the concepts of *Measurement Point*, *Stop Point* and *Checked-in PoI*.

- A *Measurement Point* is a 3-ple $(latitude,\ longitude,\ timestamp)$ collected through a sensor which indicates the presence of the user in a given point at a certain time instant.

- A *Stop Point* is a tempo-spatial cluster of *Measurement Points* which represents a geographic region where the user stopped for a while.
- A *Checked-in PoI* is a PoI, among those contained in the system knowledge base, which has been actually visited by the user.

It is worth noticing that unavoidable location errors and the presence of high-density PoIs areas can make difficult the detection of a *Checked-in PoI*, which is not always the nearest PoI to the extracted *Stop Point* .

In the following, we present the strategies adopted to tackle the *Stop Points* discovery problem, and the *Checked-in PoI* detection problem.

A. Stop Points Discovery

Several approaches have been presented in the literature in order to discover Stop Points from raw data. In particular, approaches described in [6], [7] are characterized by a good trade-off between accuracy and computational burden. Nevertheless, these approaches are characterized by some limitations that can be overcome by means of our proposal.

According to authors of [6], a *Stop Point* is discovered whenever there exist two *Measurement Points*, p_a and p_b, for which the following constraints are satisfied:

- $SpaceDistance(p_a, p_b) < \delta_d$,
- $TimeDifference(p_a, p_b) > \delta_t$,
- $TimeDifference(p_k, p_{k+1}) < \Delta_d, \ \forall k: \ a \leq k \leq b$,

where δ_d is the maximum distance a user can cover to be considered staying in the same *Stop Point*, δ_t is the minimum visiting time necessary to discover a *Stop Point*, and Δ_t is the maximum temporal distance between two consecutive samples to be considered as part of the same cluster of measurements. The main weakness of this approach is that it is not capable of dealing with possible signal losses, which can be frequent with GPS measurements are used for detecting users location.

Authors of [7] propose to solve such issue by relaxing the constraint about the maximum time threshold. *Measurement Points* that violate such constraint are stored in a temporary buffer to be compared with the next *Stop Point*. Whenever a new *Stop Point* is discovered, its distance from the previous one is evaluated; if such distance is under a given threshold, these two points are merged, together with all the *Measurement Points* stored in the buffer. Even with this improvement, the method proposed in [7] generates too many *Stop Points*, since it does not take into account any information about the *nature* of some waypoints, which often can be known a priori.

We propose to overcome such limitation by comparing the discovered *Stop Points* with information stored in a database of known waypoints, in order to merge consecutive *Stop Points* which match the same waypoint.

B. Checked-in PoI Detection

Results of the *Stop Points* discovery phase are used to identify the PoIs visited by the user, defined as *checked-in PoIs*. The easiest approach to detect a *checked-in PoI* is to select the known PoI which is nearest to the centroid of the current *Stop Point*. Such strategy, named *nearest neighbor method*, is equivalent to apply a reverse geocoder to *Stop Points* [8], [11]. Despite its simplicity, such an approach is negatively affected by unavoidable errors of GPS sensors and by the presence of areas with a high density of PoIs.

To overcome such limits, some methods have been proposed in the literature. Authors of [10] propose a spatial search system composed of two phases: a retrieval phase which extracts a set of candidate PoIs, followed by a ranking phase which produces the best raking of candidate PoIs. The retrieval phase selects the most popular venues near to user's position. The ranking of this PoIs is obtained through a supervised learning algorithm which is trained by collecting explicit feedbacks from users. The learning algorithm also includes some context information, such as the distance between *Stop Point* and PoI, the time of the day and the number of people currently checked-in at the PoI. A similar approach is proposed in [9], where the set of candidate PoIs is obtained only by verifying that their distance from the current *Stop Point* is under a given threshold. Context information used to refine the PoI ranking include PoI popularity, the number of reviews on social media, personal preferences, time of the day and weather conditions. Authors of [8] propose a generative model to detect visited PoIs from unlabeled *Stop Points*, through a Bayesian network that represents the probabilistic dependence of different factors on visited PoIs, such as user preferences for different PoI categories, duration and geographical location of the visit, PoI popularity and typical stop time.

We propose to adopt a Dynamic Bayesian Network (DBN) [17] in order to include context awareness and the knowledge about past history into the PoI detection algorithm, and in order to deal with the unavoidable noise in sensory readings and in the discovered Stop Points. In our model, *Stop Points* discovered through the enhanced method described in Sec. IV-A represent the observable manifestation of the hidden user state, i.e., the real checked-in PoI. The DBN allows to model the evolution of the hidden state over time, and the probability dependency of the current state, from past state and from context features. Moreover, differently from approaches described above, we propose to use a context information set which is not dependent on global features of PoIs, i.e. popularity or visibility on social media, but which heavily depends on the recurrent nature of human mobility [18], that can be often modeled through a weekly routine. According to this choice, our model takes simultaneously and separately into account three different aspects related to the time relation between the discovered *Stop Point* and the potential PoI: the arrival time at the *Stop Point*, the duration of the visit and the day of the week. Furthermore, we exploit the intentional nature of human mobility by considering the sequence of past visited places during the day as further contextual feature.

Fig. 2 sketches the overall structure of the PoI Detection module, and highlights the adopted DBN and its interaction with the Stop Points Discovery subsystem, which can be

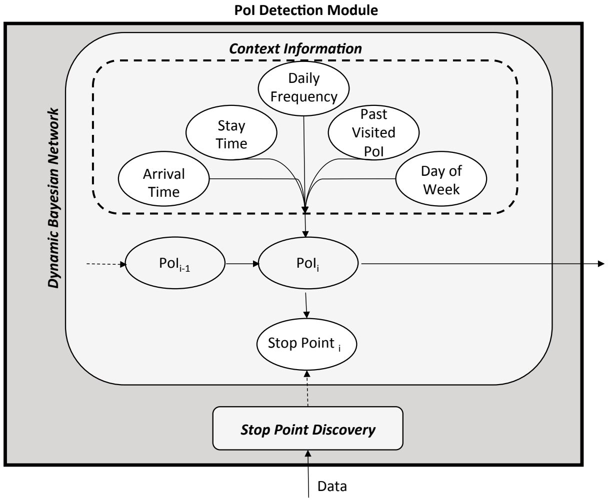

Fig. 2. Block Diagram of the *PoI Detection* module.

considered as a virtual sensor which perceives noisy manifestations of real PoIs. The main goal of this module is to infer the *i-th* PoI visited during a day, which is represented by the hidden variable x_i. The belief about such variable depends from the past history, the current sensory reading e_i, i.e. the discovered *Stop Point*, and a set of context information $c_i = (C_i^1 \dots C_i^k)$.

The characterization of the DBN requires the definition of the *sensor model* and the *state transition model*. The probability distribution $P(e_t|x_t)$ represents how *Stop Points* are affected by the current visited PoI, it is named *sensor model*, and it is inversely proportional to the distance between the centroids of the discovered *Stop Point* and the hypothesized PoI x_i. The *state transition model*, i.e., $p(x_i|x_{i-1}, c_i)$, represents the probability that the user visited a given PoI, given the previously visited PoI x_{i-1} and the current context information c_i.

The context information are obtained through a set of feature extractors, that for each user, and for each PoI class, quantify their relationship in temporal and behavioral terms. We focus on the following features: the frequency of visits during a day, the typical arrival time, the average stop time and the history of waypoints of the same class visited in the past, expressed as frequency of n-grams of past PoIs.

Since our DBN is a first-order Markov model, we define the *belief* about the PoI visited in the i-th slot, i.e. x_i as:

$$belief(x_i) = p(x_i|e_{1:t}, c_{1:t}). \quad (1)$$

As described in [16], [19], the belief can be computed with the following practical formulation:

$$belief(x_i) = \eta \cdot p(e_i|x_i) \cdot \sum_{x_{i-1}} p(x_i|x_{i-1}, c_i) \cdot belief(x_{i-1}), \quad (2)$$

where η is a normalization constant.

Since the DBN structure is known, the learning phase only aims at filling the conditional probability tables by computing

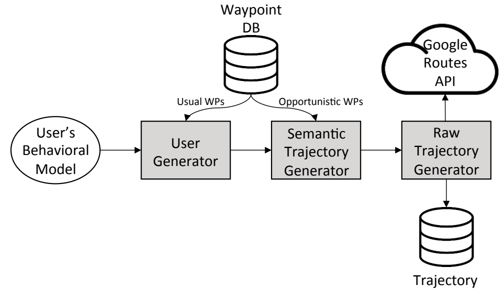

Fig. 3. Architecture of the simulation tool adopted to generate synthetic GPS trajectories.

the sample statistics for each node from the set of historical geo-referenced data.

V. Experimental evaluation

A. Dataset

In order to evaluate the effectiveness of the proposed approach, we design a simulation tool for building synthetic datasets, according to guidelines proposed in the literature [20] (see Fig. 3).

The simulation tool requires explicit knowledge of Campus PoIs, named *waypoints*, which are stored in a specific DB. Each user is represented by a set of commonly visited waypoints and a *behavior*, which is a probabilistic model for his specific mobility patterns. Such *behavior* is composed of two components: (i) a *transition scheme*, which is a Markov chain representing the transition probability from a waypoint to another, and (ii) a *pause scheme* which represents the probability distribution of stop time for each waypoint.

For each simulated user, in each day, it is generated a *semantic trajectory*, i.e., a sequence of class of waypoints. An example of semantic trajectory is the sequence $<$*Department* $\rightarrow$ *Coffee Shop* $\rightarrow$ *Library*$>$. The adoption of *semantic trajectories* to provide a high-level model of mobility behavior of users is based on the idea that humans move in order to fulfill a to-do list, which means that they move between places to switch between different activities. In our example, referred to a University Campus, after attending a lesson, a student might want to take a coffe, then go to the library.

Raw trajectories, i.e. sequences of $(lat, lng, timestamp)$ triples, are generated from semantic trajectories by exploiting *Google Routes APIs*. In particular, for each pair of consecutive waypoints, a set of position samples is generated, and then such data are corrupted by adding random noise in order to simulate the GPS errors.

Noise features can be specified as input parameters of our simulation tool. The dataset described here has been generated by adding Gaussian noise with mean value equal to 0 and standard deviation equal to 0.00005, thus to obtain some noisy readings far 10 mt from the true user's position.

The experimental evaluation presented in this work has been performed by generating a dataset of trajectories for different

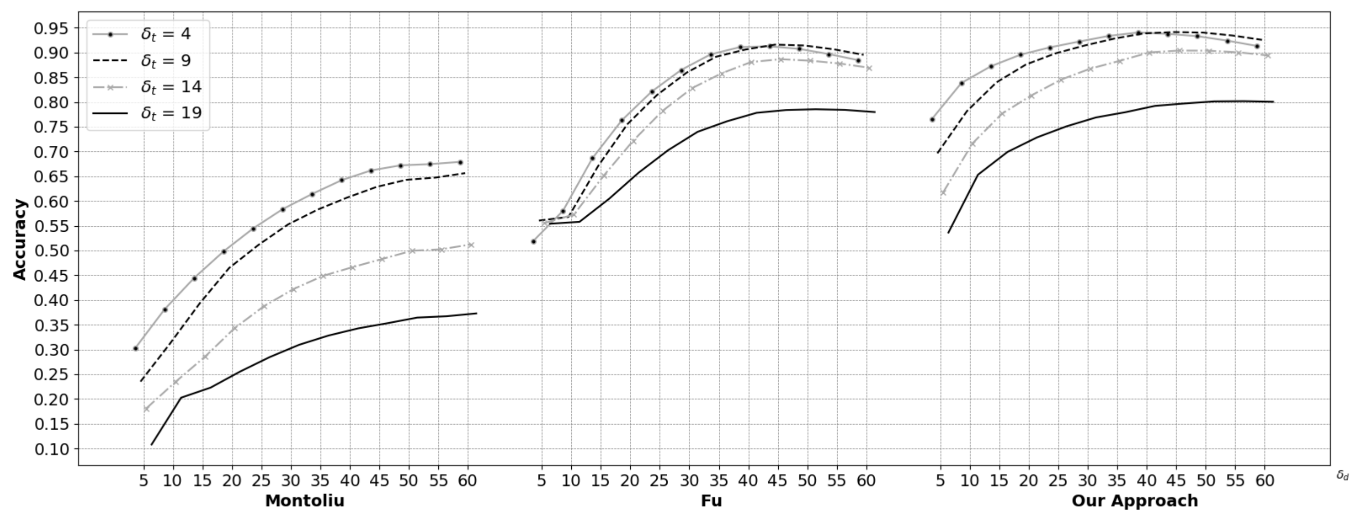

Fig. 4. Accuracy of the following *Stop Points* discovery algorithms, by varying δ_d (on the x-axis) and δ_t (grey patterns), as defined in Sec. IV-A: the approach proposed by Montoliu et al. [6], the approach proposed by Fu et al. [7], and our approach.

users, covering an academic year with a sampling rate of 5 minutes, by considering several user behavioral models.

B. Performance Metrics

The experimental results compare the performance of three different algorithms for *Stop Points* extraction, with different values for the input parameters. The first one is the algorithm proposed in [6], the second one is the technique described in [7], and the last is our proposal described in Sec. IV-A. In order to compare these approaches, the actually visited PoIs are inferred from the discovered *Stop Points* by selecting, at the end of the clustering phase, the nearest PoIs to Stop Points' centroid. As described in Sec. IV-B, the *nearest neighbor* approach presents some pitfalls, but provides an initial assessment of the techniques' effectiveness.

The overall accuracy of the proposed system can be evaluated by comparing the ground truth with the sequence of inferred PoIs, through the Damerau-Levenshtein distance [21]. Let $\Phi(\cdot,\cdot)$ be the Damerau-Levenshtein distance function between two strings, $\mathbf{PoI_G}$ be the ground-truth sequence of PoIs for a specific user, $\mathbf{PoI_D}$ be the sequence of PoIs detected by the system, and $\#\{\cdot\}$ be the function which gives the numbers of elements in a sequence, then the average accuracy for each user is computed as:

$$accuracy = \sum_{trajectories} \frac{1 - \Phi(\mathbf{PoI_G}, \mathbf{PoI_D})}{max(\#\{\mathbf{PoI_G}\}, \#\{\mathbf{PoI_D}\})}. \quad (3)$$

The division by the maximum value between the length of the real itinerary and the detected one aims to penalize strings with different lenghts, e.g., in the case of an excess of extracted *Stop Points*.

C. Experimental Results

In order to evaluate the effectiveness of our approach, we carried out two different experimental evaluations.

The first experiment compares *Stop Points* extraction methods described in Sec. IV-A, with different values for the input parameters. Fig. 4 shows the accuracy achieved by through the nearest neighbor method, under the ideal hypothesis of continuous GPS signal. This unrealistic constraint will be relaxed in the second experiment. The best performance is obtained by our approach (c), which exhibits an accuracy of 94.12%, thus increasing the accuracy of method (a) by 26.24% and of method (b) by 4%.

In the second experiment, we relax the constraint about the GPS signal continuity, which is now considered absent when a user moves inside buildings. We compare the performances of the blind nearest neighbor assignment method with the probabilistic approach for PoI detection proposed here. The training and validation of our DBN model have been performed by a 6-fold cross validation on the whole data set. Table II shows results of such comparison, and proves that the probabilistic inference, by means of different set of context features, can be useful to identify visited PoIs. Nevertheless, obtained results show that using too many context attributes can actually be detrimental to the inference accuracy, and increases the computational burden of the process. The execution times are reported in the last column of Table II: each entry is expressed as a ratio w.r.t. the highest value of execution time, i.e., the last row of the table. The features considered for this experiments are: the Arrival Time (AT) of the user at the stop point, the Stop Time duration (ST), the Daily Frequency (DF) of visits of a particular semantic class, the sequence of Past Visited Places (PVP), and the Day of the Week (DW). In order to compute DF and NG features, we have to make an approximation of the real itinerary by considering the most probable waypoint as the actually visited one.

As results show, optimal performances are achieved by considering user's arrival time (AT) and the duration of the stop (ST), obtaining nearly 7% of accuracy increase.

TABLE II
COMPARISON OF ACCURACY, UNCERTAINTY AND EXECUTION TIME FOR PROBABILISTIC POI DETECTION WITH DIFFERENT SUBSETS OF FEATURES

AT	ST	DF	PVP	DW	Accuracy	Uncertainty	Time
-	-	-	-	-	0.830	0	0.537×
✓	✗	✗	✗	✗	0.854	0.155	0.819×
✗	✓	✗	✗	✗	0.872	0.093	0.818×
✗	✗	✓	✗	✗	0.823	0.190	0.880×
✗	✗	✗	✓	✗	0.831	0.182	0.846×
✗	✗	✗	✗	✓	0.823	0.196	0.815×
✓	✓	✗	✗	✗	0.905	0.085	0.837×
✓	✗	✓	✗	✗	0.831	0.193	0.893×
✓	✗	✗	✓	✗	0.839	0.180	0.872×
✓	✗	✗	✗	✓	0.832	0.194	0.847×
✗	✓	✓	✗	✗	0.872	0.100	0.892×
✗	✓	✗	✓	✗	0.877	0.096	0.864×
✗	✓	✗	✗	✓	0.866	0.091	0.832×
✗	✗	✓	✓	✗	0.820	0.194	0.916×
✗	✗	✓	✗	✓	0.810	0.218	0.889×
✗	✗	✗	✓	✓	0.818	0.205	0.872×
✓	✓	✓	✗	✗	0.881	0.104	0.911×
✓	✓	✗	✓	✗	0.884	0.100	0.897×
✓	✓	✗	✗	✓	0.874	0.110	0.868×
✓	✗	✓	✓	✗	0.777	0.228	0.940×
✓	✗	✓	✗	✓	0.774	0.242	0.920×
✓	✗	✗	✓	✓	0.789	0.219	0.906×
✗	✓	✓	✓	✗	0.874	0.110	0.931×
✗	✓	✓	✗	✓	0.867	0.110	0.905×
✗	✓	✗	✓	✓	0.869	0.108	0.896×
✗	✗	✓	✓	✓	0.775	0.237	0.940×
✓	✓	✓	✓	✗	0.865	0.140	0.963×
✓	✓	✓	✗	✓	0.874	0.128	0.942×
✓	✓	✗	✓	✓	0.876	0.121	0.933×
✓	✗	✓	✓	✓	0.709	0.227	0.981×
✗	✓	✓	✓	✓	0.860	0.143	0.965×
✓	✓	✓	✓	✓	0.846	0.145	1.000×

VI. CONCLUSIONS AND FUTURE WORK

This paper described a system for detecting PoIs in a Smart Campus, in order to build location-aware recommender systems. Our system exploits a combination of unsupervised and supervised methods which allow to merge sensory information gathered by a pervasive sensory infrastructure, with high-level context information, through a probabilistic model. The experimental evaluation, performed on a synthetic dataset, confirm that, by considering also context information beside to sensory data, our approach has better performance than other analogous approaches proposed in the literature.

As future work, we will address other modules of the proposed context-aware recommendation system, also by including the possibility of taking into account users' activity detected from pervasive sensors as further context feature [22], [23]. Moreover, we plan to test the whole system in a real Smart Campus scenario, by including also data voluntarily shared by users. In such a scenario, we will include also techniques of reputation management [24] in order to discard feedback provided by users who intentionally send incorrect data to create a disservice.

ACKNOWLEDGEMENT

This research is partially funded by the Project VASARI, PNR 2015-2020, DD MIUR n. 2511.

REFERENCES

[1] M. C. Gonzalez, C. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–82, 2008.

[2] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani, and D. Peri. A context-aware system for ambient assisted living. In *Proc. of Int. Conf. on Ubiquitous Computing and Ambient Intelligence*, pages 426–438, 2017.

[3] F. Concone, P. Ferraro, and G. Lo Re. Towards a smart campus through participatory sensing. In *Proc. of the 2018 IEEE Int. Conf. on Smart Computing (SMARTCOMP)*, pages 393–398, 2018.

[4] V. Agate, F. Concone, and P. Ferraro. Wip: Smart services for an augmented campus. In *Proc. of the 2018 IEEE Int. Conf. on Smart Computing (SMARTCOMP)*, pages 276–278, 2018.

[5] S. Gaglio and G. Lo Re. *Advances onto the Internet of Things*, volume 260. Springer, 2014.

[6] R. Montoliu, J. Blom, and D. Gatica-Perez. Discovering places of interest in everyday life from smartphone data. *Multimedia Tools and Applications*, 62(1):179–207, 2013.

[7] Z. Fu, Z. Tian, Y. Xu, and C. Qiao. A Two-Step Clustering Approach to Extract Locations from Individual GPS Trajectory Data. *ISPRS Int. Journal of Geo-Information*, 5:166, 2016.

[8] K. Nishida, H. Toda, T. Kurashima, and Y. Suhara. Probabilistic identification of visited point-of-interest for personalized automatic check-in. In *Proc. of the 2014 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp 2014)*, pages 631–642, 2014.

[9] D. Lian and X. Xie. Learning location naming from user check-in histories. In *Proc. of the 19th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 112–121, 2011.

[10] B. Shaw, J. Shea, S. Sinha, and A. Hogue. Learning to rank for spatiotemporal search. In *Proc. of the 6th ACM Int. Conf. on Web Search and Data Mining (WSDM 2013)*, pages 717–726, 2013.

[11] X. Cao, G. Cong, and C. S. Jensen. Mining Significant Semantic Locations from GPS Data. *Proc. of the VLDB Endowment*, 3(1-2):1009–1020, 2010.

[12] D. Ashbrook and T. Starner. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. *Personal Ubiquitous Computing*, 7(5):275–286, 2003.

[13] J.G. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *Proc. of the 2nd ACM Int. Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, pages 110–118, 2004.

[14] Y. Ye, Y. Zheng, Y. K. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Proc. of IEEE Int. Conf. on Mobile Data Management*, pages 1–10, 2009.

[15] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In *Proc. of the 11th Int. Conf. on Ubiquitous Computing*, pages 21–30, 2009.

[16] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, and S. K. Das. An adaptive bayesian system for context-aware data fusion in smart environments. *IEEE Trans. on Mobile Computing*, 16(6):1502–1515, 2017.

[17] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[18] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[19] A. De Paola, P. Ferraro, S. Gaglio, and G. Lo Re. Context-awareness for multi-sensor data fusion in smart environments. In *Proc. of the Conf. of the Italian Association for Artificial Intelligence*, pages 377–391, 2016.

[20] F. Giurlanda, P. Perazzo, and G. Dini. Humsim: A privacy-oriented human mobility simulator. In *Int. Conf. on Sensor Systems and Software*, pages 61–70, 2014.

[21] G. V. Bard. Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In *Proc. of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pages 117–124, 2007.

[22] F. Concone, G. Lo Re, and M. Morana. A fog-based application for human activity recognition using personal smart devices. *ACM Trans. on Internet Technology (TOIT)*, 19(2):20, 2019.

[23] F. Concone, S. Gaglio, G. Lo Re, and M. Morana. Smartphone data analysis for human activity recognition. In *Proc. of the Conf. of the Italian Association for Artificial Intelligence*, pages 58–71, 2017.

[24] V. Agate, A. De Paola, and M. Lo Re, G.and Morana. A simulation framework for evaluating distributed reputation management systems. In *Proc. of the 13th Int. Conf. Distributed Computing and Artificial Intelligence*, pages 247–254, 2016.