ORIGINAL RESEARCH PAPER

WILEY

A multi-agent system for itinerary suggestion in smart environments

Alessandra De Paola[1,2] | **Salvatore Gaglio**[1,2] | **Andrea Giammanco**[1] |
Giuseppe Lo Re[1,2] | **Marco Morana**[1,2] [iD]

[1]Università degli studi di Palermo, Palermo, Italy

[2]Smart Cities and Communities National Lab CINI - Consorzio Interuniversitario Nazionale per l'Informatica, Palermo, Italy

Correspondence

Marco Morana, Università degli studi di Palermo, Palermo 90128, Italy.
Email: marco.morana@unipa.it

Abstract

Modern smart environments pose several challenges, among which the design of intelligent algorithms aimed to assist the users. When a variety of points of interest are available, for instance, trajectory recommendations are needed to suggest users the most suitable itineraries based on their interests and contextual constraints. Unfortunately, in many cases, these interests must be explicitly requested and their lack causes the so-called cold-start problem. Moreover, lengthy travelling distances and excessive crowdedness of specific points of interest make itinerary planning more difficult. To address these aspects, a multi-agent itinerary suggestion system that aims at assisting the users in an online and collaborative way is proposed. A profiling agent is responsible for the detection of groups of users whose movements are characterised by similar semantic, spatial and temporal features; then, a recommendation agent leverages contextual information and dynamically associates the current user with the trajectory clusters according to a Multi-Armed Bandit policy. Framing the trajectory recommendation as a reinforcement learning problem permits to provide high-quality suggestions while avoiding both cold-start and preference elicitation issues. The effectiveness of the approach is demonstrated by some deployments in real-life scenarios, such as smart campuses and theme parks.

1 | INTRODUCTION

Creating effective itineraries between points of interest (PoIs) represents a difficult task for users, given the quantity and the variety of constraints to be considered. In general, human mobility is *intentional* [1], since people move from one place to another in order to fulfil a real-life goal, such as borrowing a book in a library, or seeing an artwork in a museum. Therefore, in the design of an intelligent system capable of suggesting trajectories, the first constraint to be considered is the semantic of the PoIs and the people's movements: realistic trajectories should allow the inclusion of multiple and different semantic classes of PoIs. Furthermore, users may express different personal preferences for PoIs belonging to the same semantic

class. If users do not have much time available, for instance, their choice will be biased towards trajectories with the minimum travelling distance or queueing times among PoIs. Moreover, a user moving in an unknown area may find it difficult to proceed among a multitude of PoIs. All these aspects point out the need for automatic tools capable of filtering several information and suggesting trajectories aligned with semantic, temporal and spatial constraints. In this context, Recommender Systems (RSs) play a fundamental role in maximising users' satisfaction in situations where multiple constraints may also be subject to real-time factors [2]. In particular, *smart environments*, where measurements gathered by a pervasive sensory infrastructure [3] are used to provide intelligent services to users [4], represent a significant scenario

which may benefit from properly designed trajectory RSs. *Smart museums*, for instance, are characterised by customers wishing to spend their visit time for their preferred masterpieces, and conveniently postpone the contemplation of currently over-crowded artworks. Similarly, in a *smart campus* [5], it is useful to track libraries with free seats or parking spaces with available slots which can be included in the recommendations. Among the most effective recommendation algorithms, *collaborative filtering* approaches are based on collecting and analysing a large amount of information on users' preferences in order to build a model representing their past behaviour, and predict interesting items for the current users based on their similarity with other ones [2]. In the context of trajectory recommendation, this paradigm assures that suggested trajectories have always been travelled before by other users, thus increasing the reliability of recommendations. However, many approaches for suggesting trajectories show some drawbacks, ranging from the lack of considering semantic information not explicitly related to geographic coordinates, to the need for collecting user preferences explicitly. Moreover, traditional *collaborative filtering* approaches present an entry barrier in terms of users' past interactions needed to make accurate predictions. This is known as the *cold start* problem, which arises when a new user, without any previous history, requests a recommendation [6]. Nevertheless, supporting the itinerary decisions of unknown users is especially important in scenarios where users have only few interactions with the RS, for instance, the visitors of a museum. An emerging way [7–9] to deal with this challenging problem lies in adopting a *reinforcement learning* strategy, considering the two-faced aim of a *RS*: on the one hand, it aims to maximise user satisfaction with the immediate itinerary suggestion (*exploitation* of known knowledge), on the other hand, it wants to gain new knowledge about user preferences (*exploration* of new information) to improve long-term reward, although with some effect on short-term satisfaction. This dilemma is typically conjugated in the context of *RSs* as a *Multi-Armed Bandit* problem [7, 10]. According to this model, using an arm of an imaginary row of slot machines corresponds to selecting an action according to a specific policy. The player (i.e. the RS) has to select which arm to use, and that choice produces the element to suggest.

This work proposes a collaborative and context-aware intelligent system for trajectory recommendation. It is based on a multi-agent architecture that allows to collect and progressively exploit user feedbacks, following a *reinforcement learning* approach. Semantic, spatial and temporal features of users' movement behaviours are extracted by a profiling agent. A clustering algorithm aggregates users with similar behaviours, originating sets of trajectories that share characteristic traits. A recommendation agent analyses contextual information and current user's past interactions in order to establish its most significant coupling with one of the clusters of trajectories, following a Multi-Armed Bandit policy, and to provide the most appropriate suggestions. Such coupling is dynamically updated accordingly to user feedback, so that the system can refine its recommendations after new knowledge acquisition.

We have validated our approach on a dataset of mobility traces of users moving in a *smart campus* [11, 12] and on five public datasets of users visiting attractions in theme parks [13]. The experimental results show that our system provides accurate recommendations, and that the proposed profiling method can correctly identify separate classes of movement behaviour.

The remainder of the paper is organised as follows: Section 2 discusses related work in human mobility profiling and trajectories recommendation. Section 3 outlines the proposed system architecture, discussing the features and the clustering methodology to extract groups of similar users from the data, and describing the Multi-Armed Bandit framework beneath the recommendation agent. Section 4 describes the experimental setting together with the metrics used to evaluate the performance of the intelligent system and analyses our experimental results. Conclusions are presented in Section 5.

2 | RELATED WORK

In recent years, the proliferation of location-aware technologies, caused by the widespread use of personal smart devices, has allowed a vast amount of studies aiming at understanding various aspects of human mobility. In what follows, some of these studies will be discussed, highlighting the mobility features considered in turn.

A first simple approach for defining groups of people with similar movement behaviours entails the comparison of the single places they visit. In [14], the authors cluster foursquare check-ins by means of *topic modelling*: users who visit PoIs falling into the same topic are grouped together. This approach does not take into account the concept of trajectory, as it neglects the sequential nature of users' movements. Similarly, in [15], the information on visits distribution is interpreted in a probabilistic manner for building a user mobility profile, which is then used for predicting only the next place to visit.

The research study in [16] defines a *mobility index* by combining the frequency and geographical range of movements in order to capture temporal and spatial aspects of mobility behaviours. This index implies the calculation of several features such as the number of trips, the number of unique regions where users lingered for a significant amount of time (also called *stay points*), the specific sequences of visited PoIs, the total distance and travel time. This rich feature set misses more specific considerations on the temporal aspects of mobility, such as its weekly recurrent nature [19].

The authors of [17] characterise human mobility by selecting features as the amount of movement, the distributions of visits to PoIs, the entropy of a user's location history and its degree of predictability. We opted for the adoption of the entropy in our work as the characteristic feature, in order to provide a measure of the uncertainty of users' movements.

In [18], the authors use an event-based algorithm to cluster individuals according to their mobility behaviour. They analyse the periods of presence/absence in particular regions by imposing 6 classification rules, which consider the possible sequence of events during each day of the week. This work

introduces the important concept of differentiating users' behaviours based on their weekly mobility patterns, which we also exploit in our work by computing the features separately for each day of the week.

The authors of [8] studied the discriminative power of several features in distinguishing between different datasets of human mobility traces. The features considered include as follows: the convex hull enclosing all users positions, the entropy of individual trajectories and the degrees of freedom required to describe a particular trajectory. Although the convex hull represents an interesting point of view on the spatial representation of trajectories, it goes beyond the scope of this work, where we focus on trajectories settled in a single *smart environment* with a modest quantity of PoIs (see Section 4.1), thus, multiple trajectories would share the same convex hull.

Our work borrows a mixture of the features proposed in the described studies. Research studies have shown that content information on the visited PoIs provides important contributions to the inference of personal interests, for example a user who visits a PoI with semantic tag 'vegetarian restaurant' gives information on his diet [19]. This is the reason why we add the sequences of visited semantic classes of PoIs as a mobility feature. Moreover, users' activities show strong temporal cyclic patterns in terms of hour of the day and day of the week [20, 21], therefore, we also take into account the arrival time in PoIs semantic classes, and we compute our set of features separately for each day of the week. In this way, the clustering algorithm will take into account the semantic, spatial and temporal behaviour similarity in the different weekdays. The set of features we adopt is discussed in detail in Section. 3.1. A tabular representation of the features considered in the discussed studies is presented in Table 1. The motivations behind the features chosen for this study will be described in Section 3.1, together with the definition of the feature set.

Several surveys provide a comprehensive overview on the trajectory recommendation problem in the tourism domain. The authors of [22] present a set of relevant recommendation paradigms for this scenario by highlighting the type of service offered to the end user, for example suggesting hotels available in a certain period of the year, or finding a specific combination of PoIs reachable within a certain time range. In [23], different approaches that face recommendation as an optimisation problem with multiple constraints dictated by user needs are discussed. In [24], the authors provide multiple points of view on the recommendation problem in the tourism domain, addressing the different methods to gather information about tourists' behaviour, and the composition of a coherent tour both for individuals or groups.

Since a trajectory is a concatenation of visited PoIs, a related research area to trajectory recommendation is the *top-k* PoIs recommendation [25, 26]. These approaches perform a variant of the *matrix factorisation* algorithm which leverages several context information [13]. Unfortunately, as shown in [27], the transition probability between different PoIs is non-uniform, meaning that venue visits are not independent of each other. Thus, the sequential nature of a trajectory has to be taken into account in order to perform effective recommendations. In [28], the proposed system expects a visit sequence of semantic PoIs' categories as input in order to match specific geographic routes, resulting in an onerous interaction request from the user. In [13], geo-referenced photos taken in theme parks are used to infer queueing times at attractions, and trajectories with the least amount of waiting time are recommended. The system also maximises user interests scores and attraction popularity. However, this approach requires starting and ending PoIs as input which may not always be known a priori, and heavily suffers from the cold start problem, needing a tree of visit counts in order to select the next PoI in the trajectory. In [29], the set of potentially interesting PoIs for the user are arranged in a graph structure where edges are weighted with the shortest route travel time, and then, a score function is used to filter the paths according to context and users' preferences related to PoIs categories. But users' preferences are collected by means of a form proposed during the registration phase, and a query to the recommender system has to include explicit parameters such as the starting and destination point or the maximum number of PoIs to visit, thus reducing the automation of the approach. All trajectory recommendation systems discussed so far are not *collaborative*

T A B L E 1 Summary of the features considered in the discussed studies on human mobility profiling

	Features													
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
[14, 15]							✓							
[16]	✓	✓	✓	✓	✓									
[17]			✓				✓	✓	✓					
[18]							✓							✓
[8]								✓				✓	✓	
Our approach		✓	✓	✓	✓	✓	✓	✓		✓	✓			✓

Note: These features are as follows: (1) number of trips; (2) number of unique stay points; (3) radius of gyration; (4) travel time; (5) sequence of visited PoIs; (6) sequence of visited PoIs semantic flags; (7) visits distribution; (8) entropy of user history; (9) predictability of user history; (10) arrival time for PoIs semantic labels; (11) average stay time for PoIs semantic labels; (12) convex hull; (13) degrees of freedom; and (14) features computed on a weekly basis.

TABLE 2 Notations for the MAB problem formulation

Notation	Description
t	Round
T	Horizon
$\mathcal{C}$	Set of K clusters (arms)
c_k	k-th arm
A_t	Arm pulled (cluster selected) in round t
η_{c_k}	Number of times in which the cluster c_k has been selected
z_t	Recommended trajectory in round t, extracted from the selected cluster A_t
r_{A_t}	Reward by pulling the given arm in the $t-th$ round
Ξ_t	Sequence of past interactions $<A_{t-1}, r_{A_{t-1}}>$ observed until round t
π	Policy for sequential trajectories recommendation
μ_{c_k}	Experimental mean for arm c_k
$\sigma_{T,\pi}$	Cumulative *Reward* after T rounds following policy π
$\rho_{T,\pi}$	Cumulative *Regret* after T rounds following policy π

filtering-based systems, therefore, trajectories never travelled before could be suggested to users.

In Ref. [13], geo-tagged photos are also exploited by system in [30], where visual features (such as the colour histogram and other vectorial representations coming from SIFT descriptors) are merged into the *matrix factorisation* model, allowing a better management of the *cold start* problem. However, the approach requires a simultaneous use of multiple machine learning algorithms to process photos' contents, which can be computationally demanding. In [31], a first pre-processing phase aims at learning vectorial representations of PoIs from textual information retrieved from Wikipedia. These textual information are merged with user-PoIs visit relationships and PoIs semantic classes in order to compute a profit score for each PoI, and a variant of the *travelling salesman problem* is adopted to build a trajectory. However, this approach heavily suffers from the user cold-start problem, being unable to recommend trajectories to users with no previous visiting history.

Among the recommendation approaches which leverage clustering algorithms to infer common traits in user behaviours, [32] adopts the k-means algorithm in order to group similar users and suggest elements according to the cluster they belong. Such an approach requires each user to belong to only one group; conversely, in our approach, the cluster of belonging may change over time, according to the online feedback received by the user. The method in [33] provides a different perspective, performing a hierarchical clustering algorithm to aggregate PoIs in order to identify a path of maximum interest for the current user. However, this approach requires the specification of the initial and final PoIs, which may not always been known in advance, especially when visiting an unknown environment. The AGREE model [34] adopts a bidirectional recurrent neural network in order to infer affinity between users' preferences and provide a route suggestion; however, even this approach is not capable of dynamically associating users with groups as new feedback responses are provided. The authors of [35] propose a system to extract relevant features from call for tenders, leveraging the k-means algorithm to infer and suggest the most suitable call for the particular skill set of a given company. However, this clustering approach requires an a priori specification of the number of clusters, making it difficult to set up an online recommender system. In [36], a collaborative neighbourhood-based approach for movie recommendations is proposed. The recommendations exploit the k-means algorithm, which enables the identification of groups of similar users, and the Mahalanobis distance in order to compute the similarities of users belonging to the same cluster. Even this approach fails in accounting for the variable nature of the association of a user to a given cluster, which is an aspect of utmost importance in the design of a dynamic system. The work in [37] adopts the k-medoids algorithm to infer the neighbourhoods of users with shared preferences, as well as items with similar features, and tags labels underlying the same interests. The identified clusters are then refined based on friend relationships retrieved from social media. However, the approach neglects the possibility for users to change their interests over time.

In the broader research field of *Recommender Systems*, our proposal relates to the *collaborative filtering bandit* subfield, where existing literature proposes different design choices for modelling the *recommendation* problem in terms of *arms* to *explore* or *exploit*. In a pioneering work [10], all the individual elements of the application domain are considered directly as *arms*, their attributes are used as context information to guide future suggestions. A different modelling choice entails considering users as *arms* which

FIGURE 1 Overview of the *collaborative filtering bandit* system for trajectory recommendation
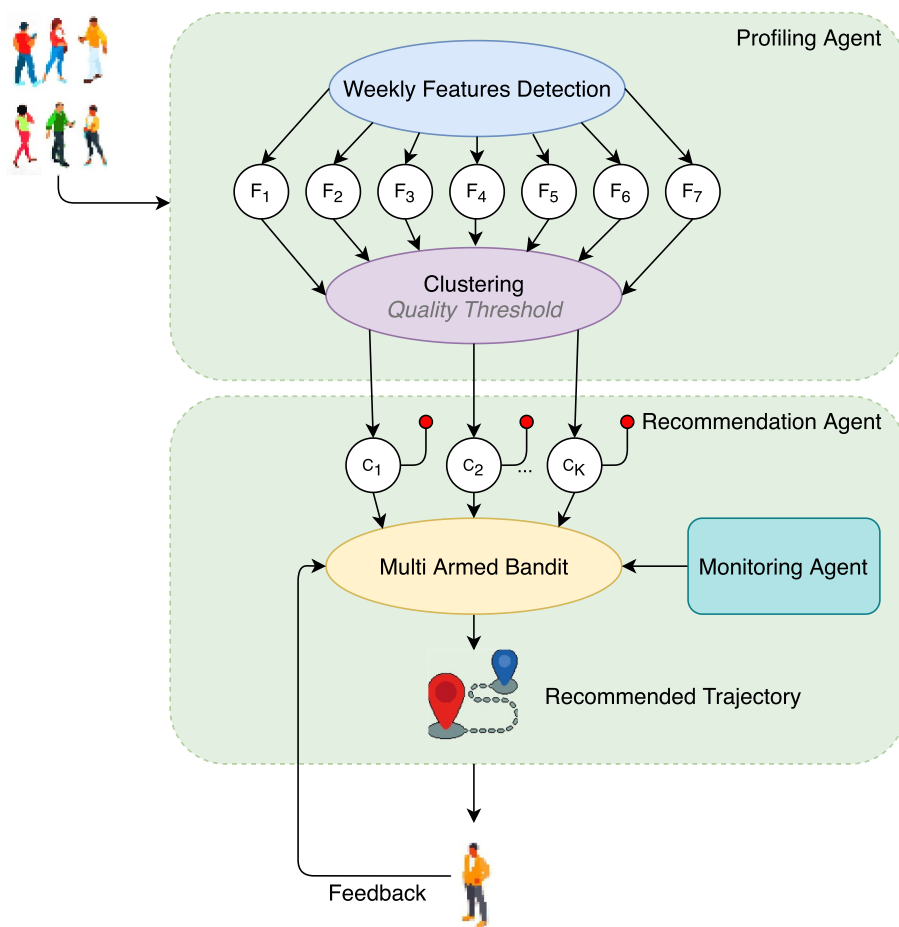
will be chosen as potential neighbours [38], this choice allows to add uncertainty in the classic neighbourhood-based collaborative filtering scheme.

Considering the above review of the state-of-the-art literature, this study proposes a context-aware collaborative system for trajectory recommendation in smart environments equipped with a pervasive structure of sensors for position detection, such as *smart campus*, *museum* or *theme park*. Differently from earlier works, our system focusses on grouping similar users by means of a weekly set of semantic, spatial and temporal features, and dynamically finding the best pair < user, cluster > by means of the online feedback against the recommendation.

3 | MULTI-AGENT RECOMMENDATION SYSTEM

The aim of our system is the recommendation of trajectories to users in an online and collaborative mode, exploiting a cloud architecture. The system reckons on two agents addressing logically distinct phases, as shown in Figure 1. The aim of the *Profiling Agent* (Sec. 3.1) is two-fold: it deals with the detection of a weekly set of features encoding semantic, spatial and temporal behaviours; it then performs a clustering algorithm to

extract groups of similar users. The extracted clusters represent trajectories which share latent parameters mirroring the behaviours of the grouped users. The *Recommendation Agent* (Sec. 3.2) exploits the clusters as sources of recommendations, and suggests a trajectory according to the occurrence frequencies in the chosen cluster. A *reinforcement learning* sub-agent (implementing a *Multi-Armed Bandit* policy) is responsible for the choice of the best-fitting cluster for the behaviour of the current user, whose selection is further refined by means of the received feedback. Context information regarding the crowdedness of the PoIs, and the travelling distances between them is integrated by a *Monitoring Infrastructure* sub-agent, which is responsible for modifying the probabilities of trajectories extraction according to the data received from physical sensors, for example wireless sensor networks (WSNs) [39], body area networks [40], or other pervasive data gathering technologies related to ambient intelligence [41] and crowdsensing methods [42].

3.1 | Profiling agent

The first step of the *Profiling Agent* is the identification of different groups of users according to semantic, spatial and temporal features. Since human mobility can be modelled as a
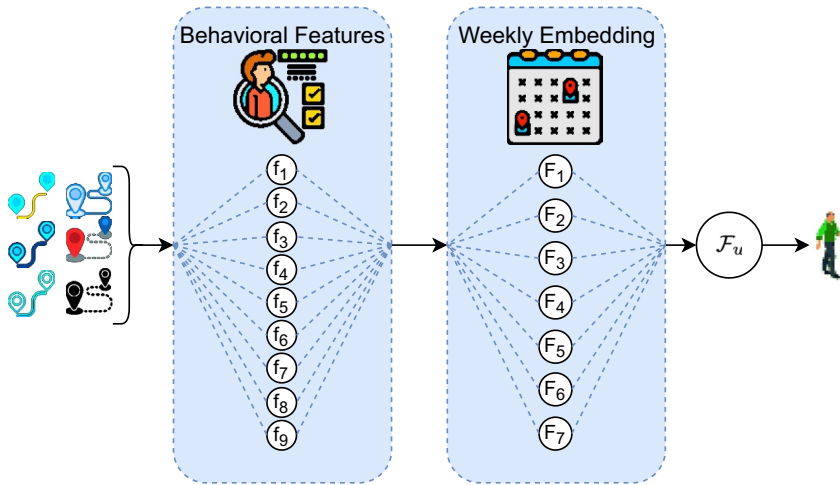

periodic routine, for example on a weekly basis [19], it is profitable to track separately the behavioural patterns for each day of the week. Therefore, in order to formulate a vectorial representation of user's movements, we chose to adopt a set of 9 features which are computed independently for each day (see Figure 2).

The vectorial representation of a given user u, $\mathcal{F}_u$ is defined as follows:

$$\mathcal{F}_u = [F_1, F_2, F_3, F_4, F_5, F_6, F_7], \tag{1}$$

where each F_d is the vector of features computed for the d-th weekday. Each F_d is composed of a set of 9 features:

$$F_d = [f_1, \ldots, f_9]. \tag{2}$$

Generally, PoIs have a semantic tag which refers to the intended functionality and the underlying activity that is to be performed, such as park, school, library [1], or associated to the type of available resources [43]. For this reason, trajectories between PoIs can be interpreted in two ways: as a sequence of integers representing the identifiers of the PoIs, for example $PoI\ 32 \to PoI\ 54$; or as a sequence of the semantic tags associated with the PoIs, for instance $Library \to Park$. The latter reflects the concept of *semantic trajectory*, first proposed in [44]. The generic i-th trajectory t_i is thus composed of the sequence of X visited PoIs: $t_d = \{x_1, \ldots, x_X\}$. Some of the behavioural features we adopt are computed separately for each semantic class of PoIs. In such cases, we denote with ϕ_j the single feature value computed for the j-th semantic class, and then, with

$$f_q = [\phi_1, \ldots, \phi_s], \tag{3}$$

the comprehensive value of the q-th feature for the s PoIs semantic classes contained in the considered dataset.

As described in Eq. 2, we adopted nine features which are now formally defined together with the motivation behind their choice:

- $f_1 \to$ frequency of visits for each semantic class: let ϕ_j be the frequency of visits for the j-th semantic class, which is defined as the ratio

$$\phi_j = \frac{\#\ trajectories\ containing\ j-th\ class}{\#\ trajectories}, \tag{4}$$

The feature f_1 is aimed to collect information about all the PoIs, $f_1 = [\phi_1, \ldots, \phi_s]$.

The frequency of visits for every semantic class allows to characterise the activities performed by the users, according to the PoIs they have visited most often.

- $f_2 \to$ PoI with longest permanence time: it is defined as the couple

$$f_2 = [x, y], \tag{5}$$

where x is the unique identifier of the PoI, and y is its semantic class. This feature emphasises the predilection towards a specific PoI as a means to assess the similarity between users.

- $f_3 \to$ average permanence time for each semantic class: the stay time in each PoI category is averaged against all visits, thus the single ϕ_j for the $j-th$ semantic class is computed as follows:

$$\phi_j = \frac{\sum stay\ time\ @\ j-th\ semantic\ class}{\#\ visits\ to\ j-th\ semantic\ class}, \tag{6}$$

and the comprehensive value of the feature is the set $f_3 = [\phi_1, \ldots, \phi_s]$.

The average stay time for PoIs semantic labels enables a temporal categorisation of users based on which places, and which activities, they spend most of their time.

- $f_4 \rightarrow$ typical arrival time for each semantic class: we decomposed a day into 48 time slots of 30 min each. We then considered the three most frequent arrival times ϕ_{jk} for each $j - th$ semantic class, with $k \in [1, 2, 3]$, thus, the comprehensive feature is composed of the following:

$$f_4 = [[\phi_{11}, \phi_{12}, \phi_{13}], \ldots, [\phi_{n1}, \phi_{n2}, \phi_{n3}]], \qquad (7)$$

The arrival time for PoIs semantic labels allows to distinguish users based on which portion of the day they prefer to conduct their trips.

- $f_5 \rightarrow$ average travel time: it is defined as the ratio

$$f_5 = \frac{\sum_i \sum_j stay\ time\ @\ j - th\ PoI \in i}{\#\ trajectories}, \qquad (8)$$

where i represents the $i - th$ trajectory of the current user, and j the $j - th$ PoI in trajectory i.

The travel time enables a categorisation of users based on the quantity of time they invest in their trips, which is an effective feature to identify users who travel for work or leisure, and in the latter case, a means to identify people taking small breaks or longer ones with their family.

- $f_6 \rightarrow$ radius of gyration: it is defined as the characteristic distance travelled by a user, centred in the trajectory's centre of mass [45]. Let N_i be the number of samples (e.g. GPS (lat, lon) coordinates) recorded for the $i - th$ trajectory, and M_{ud} be the number of trajectories for the user u in day d, the computation of the radius of gyration is as follows:

$$f_6 = \frac{\sum_{\psi=1}^{M_{ud}} \sqrt{\frac{1}{N_i} \sum_{n=1}^{N_i} (s_n - s_{cm})^2}}{M_{ud}}, \qquad (9)$$

where $s_n = (lat_n, lon_n)$ are the coordinates of the $n - th$ sample, and $s_{cm} = (lat_{cm}, lon_{cm})$ are the coordinates of the centre of mass. If the dataset under examination does not contain a complete set of position measurements for the users, this feature is computed by considering only the coordinates of the visited PoIs in trajectories.

The radius of gyration provides information on the area swept by the user's movements, allowing to distinguish between persons accustomed to travelling long or short distances.

- $f_7 \rightarrow$ entropy of movement: it measures the degree of predictability of a user [19]. Let $t_i = \{x_1, \ldots, x_X\}$ be the sequence of X PoIs visited by a user in the $i - th$ trajectory, the entropy is computed as follows:

$$f_7 = -\sum_{t'_i \subset t_i} p(t'_i) \cdot log_2 p(t'_i), \qquad (10)$$

where $p(t'_i)$ is the probability of finding a particular time-ordered pair, t'_i, in trajectory t_i, that is $t'_i \in \{\{x_1, x_2\}, \ldots, \{x_{X-1}, x_X\}\}$. An entropy value $f_7 = 0$ implies extreme regularity and predictability of movements.

The entropy of user history permits to distinguish between users who exhibit a methodical behaviour from those whose movements are less predictable.

- $f_8 \rightarrow$ semantic trajectories: it is defined as the user's history of visited PoIs semantic classes sequences. Each PoI is associated with a specific semantic label which provides information about the nature of the activity that can be performed. The $i - th$ semantic trajectory st_i consists of the sequence of Y semantic classes associated to the PoIs visited in a day:

$$st_i = [y_1, \ldots, y_Y]. \qquad (11)$$

Hence, the history of semantic trajectories is defined as follows:

$$f_8 = [st_1, \ldots, st_T], \qquad (12)$$

where T is the total number of trajectories travelled by the current user.

Reasoning upon the visited semantic classes allows to discriminate users according to the activities they enjoy during their trips.

- $f_9 \rightarrow$ PoI ids trajectories: it is defined as the user's history of visited PoIs ids' sequences. The generic $i - th$ trajectory t_i is composed of the sequence of X visited PoIs:

$$t_i = \{x_1, \ldots, x_X\}. \qquad (13)$$

Then, the feature f_9 is defined as follows:

$$f_9 = [t_1, \ldots, t_T], \qquad (14)$$

where T is the total number of trajectories followed by the user. This feature complements f_8 and allows to obtain more precise information to evaluate the similarities between users, analysing not only the activities but also the specific locations and sequence of visited PoIs.

The *Profiling Agent* is also aimed to identify the user groups with similar habits. First, we use a min-max scaling approach to project features into the range $[0, 1]$. However, not all features have an algebraic interpretation, and an alternative distance measure has to be defined in order to perform clustering. In particular, for feature f_2 (PoI with longest permanence time), we assign the minimum distance value 0 when the

PoIs are identical, an intermediate value of 0.25 when they share the semantic flag, and the maximum value 1 when they are totally different. We adopted the *Damerau Levenshtein distance* [46] $DL(a, b)$ to compute the distance between trajectories. It is generally used to calculate the distance between two strings, and corresponds to the number of steps needed for converting string a into string b by adding, deleting, substituting single characters, or transposing two adjacent characters. In order to obtain a value in the range $[0, 1]$, we then consider the ratio:

$$Dist(a, b) = \frac{DL(a, b)}{Max(Length(a), Length(b))} \quad (15)$$

to be the actual distance between two strings, where $DL(\cdot)$ is the *Damerau Levenshtein distance*, and $Length(\cdot)$ denotes the number of characters in the string. In the scenario addressed here, we consider each single PoI in a trajectory as a single character in a string, therefore, the measure $Dist(a, b)$ discussed above can be used to assess the distance between two trajectories a and b. This formulation allows to compare the behaviours encoded by features f_8 and f_9. In particular, for these features, we consider the histories of trajectories H_{u_i}, H_{u_j} of users u_i and u_j as sets, then, we consider the *Cartesian product* $H_i \times H_j$ and compute the average *DL distance* between all its couples in order to evaluate the distance between the users. A similar reasoning can be done for the f_4 feature, where each ϕ_{jk} is interpreted as a single character in a string.

Once the features $\mathcal{F}_u$ have been extracted for each user u in the dataset, the *Quality Threshold Clustering* (QTC) algorithm [47] is used in order to group together similar users. This choice is motivated by its feature of not requiring the a priori specification of the number of clusters to be found. Indeed, elements are progressively grouped while maintaining the *quality* of each cluster above a certain threshold, in terms of *intra-cluster* distances. In order to compute such distances, we adopted a compound distance measure, with the peculiarities for features f_2, f_4, f_8, f_9 above discussed. QTC requires the choice of two parameters, namely the maximum cluster diameter d_{max}, and the minimum number of users m_u a cluster has to contain. In this work, we automatically compute these parameters as follows: let U be the number of users with at least one trajectory in the dataset, and let $\Omega \in \mathcal{R}^{UxU}$ be the matrix of the inter-user feature distances, we define $m = \frac{3 \cdot U}{100}$ as the 3% of users in the dataset and $d = mean(\Omega)$ as matrix Ω mean value. The value of m depends on the minimum number of elements needed to create a cluster; in our smart campus scenario, for instance, we chose to consider clusters composed of at least 5 users, so as to observe a small variance in the trajectories extracted from each cluster. This choice corresponded to set m as the 3% of the users available in the dataset. We measured the effectiveness of such choices against a synthetic dataset of mobility traces, which exhibits the ground truth labels of the users belonging to specific groups (see Section 4.1).
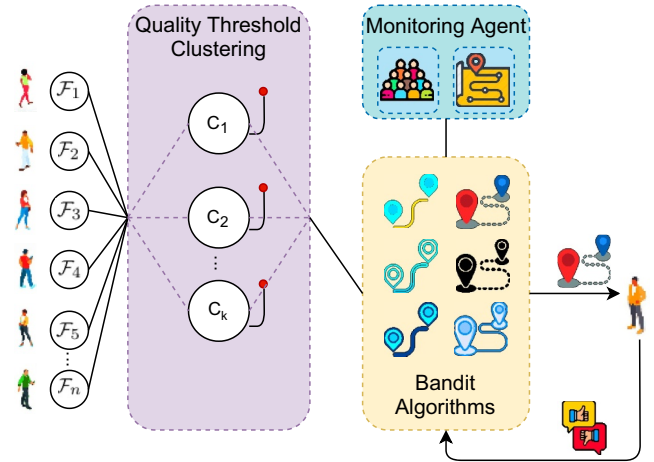

FIGURE 3 Users' clusters are interpreted as arms in a MAB fashion. An arm pull corresponds to the extraction of a trajectory from the corresponding cluster. A monitoring infrastructure integrates information about the current user context

3.2 | Recommendation agent

The *Recommendation Agent* is responsible for providing trajectories' recommendations aligned with users' preferences, which are learnt in real time by means of the received feedback. The main idea is to consider each cluster identified by the *Profiling Agent* as an *arm* in the Multi-Armed Bandit (MAB) paradigm (see Figure 3), where a *policy* infers the best pair *< user, cluster >* and modifies future recommendations accordingly.

The *MAB problem* (see notation in Table 2) deals with the tradeoff between the *exploration* of new actions to accomplish a goal, which involves the accumulation of new information that could increase profit in the long term, and the *exploitation* of the empirical best action as soon as possible, thus maximising the immediate profit [48]. In the traditional formulation, a *k-Armed Bandit* problem is composed of k slot machines (also called 'one-armed' bandits or *arms*) with their own reward probability distribution. A gambler tries to maximise, within a limited number of T rounds, the total obtainable profit by *pulling* the arm deemed most profitable in each round. The *pulling* process concludes with a *reward* returned by the selected slot machine.

In the context of our work, being $\mathcal{C} = \{c_1, c_2, ..., c_K\}$ the set of K users' clusters identified by the *Profiling Agent*, these are interpreted as the K *arms* to pull in order to obtain a reward. The gambler is represented by an *agent* who chooses an action $A_t \in \mathcal{C}$ for each round $t \in \{1, ..., T\}$. Choosing an action A_t implies the extraction of a trajectory z_t from those grouped in the selected cluster, according to their frequency of occurrence. The current user u returns a *reward* (feedback) r_{A_t} in response to the recommendation of trajectory z_t from cluster $A_t \equiv c_k$. This feedback helps the agent in refining the selection of the best pair $< u, c_k >$. Indeed, the choice of A_t depends on the user *history*, which is defined as the set of couples $< action, reward >$ observed in the past, that is $\Xi_t = \{< A_1, r_{A_1} >, ..., < A_{t-1}, r_{A_{t-1}} >\}$. The *arm* selection

TABLE 3 QTC evaluation metrics against our UniPa Smart Campus dataset

| ARI	AMI	HO	CO	VM
0.319	0.343	0.446	0.429	0.438

strategy is governed by an *agent*, embodied by a *MAB policy*, which is a mapping π from history to actions, that is $\pi : \Xi_t \to A_t, \forall t \in [1, T]$. For each *arm*, the *policy* tracks the *experimental mean*, which is defined as the arithmetical mean of the observed pulling *reward*: $\mu_{c_k} = \frac{\sum_{t=1}^{T} r_{A_t}}{\eta_{c_k}}$ with $A_t \equiv c_k$, where η_{c_k} is the number of times in which the $k - th$ arm has been selected. The *experimental means* are estimates of the *arms'* true reward probability distribution: more selections of the same *arm* refine its reward probability estimate. The effectiveness of an agent's selection policy is measured both in terms of its cumulative *reward* and *regret*. Given a policy π, the cumulative *reward* after T rounds is defined as $\sigma_{T,\pi} = \sum_{t=1}^{T} r_{A_t}$, and it represents the total profit gained by the policy following its choices. In contrast, the cumulative *regret* measures the loss in the total *reward* due to the execution of a non-ideal *policy* π, and it is defined as $\rho_{T,\pi} = \sum_{t=1}^{T} r_{A_*} - \sum_{t=1}^{T} r_{A_t}$, where A_* corresponds to the actual user's cluster.

In our trajectory recommendation setting, in each round t, a *policy* draws a trajectory to suggest z_t from a certain cluster of users c_k. The trajectory z_t is drawn in a probabilistic manner by considering the frequencies of occurrence in c_k as probabilities of being selected. In order to perform an offline evaluation of the approach, we adopted the already mentioned *Damerau-Levenshtein distance* [46]: we compare the recommended z_t with the most ξ similar trajectories in the current user history Ξ_t, where ξ is a tunable parameter, for example we used $\xi = 5$ because five of the six datasets we considered have an average number of trajectories for users less than 5.

An online deployment of the system may interactively benefit from the real-time user feedback about z_t. The system can be used to recommend both semantic or PoIs ids trajectories (see Section 3.1). The trajectories' selection probabilities are influenced from the Monitoring Infrastructure in the following way: the extraction probability of each trajectory is multiplied for the reciprocal of its visited PoIs' crowdedness (i.e. the number of people currently present in the PoI). Then, considering the current position of the user, a distance-based ranking of the trajectories is imposed by multiplying their extraction probabilities for the reciprocal of the total travel distance covered. Indeed, users tend to visit nearby destinations before visiting farther destinations when the semantic meaning behind places is equal [49]. This context integration mechanism is depicted in Figure 4.

For the cluster selection phase, we tested several bandit policies in order to empirically assess which is the best option for every dataset considered:
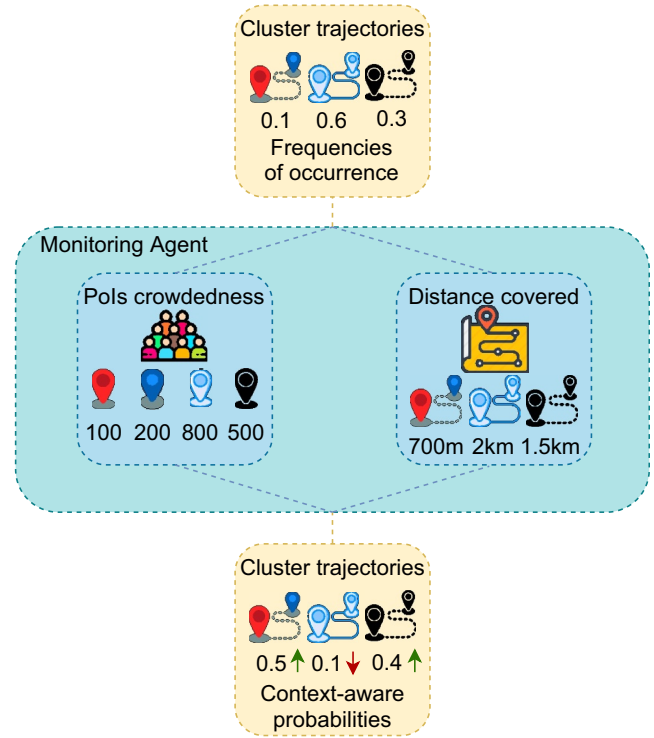
FIGURE 4 The Monitoring Infrastructure agent is responsible for the integration of context information: PoIs' crowdedness and the trajectories' covered distances are used to refine the extraction probabilities of the trajectories in the selected cluster, and suggest less congested trajectories

- *Random*: a random cluster is selected in each round.
- *ϵ-greedy* [48]: a random cluster is selected with probability ϵ, whereas the cluster c^* with the highest *experimental mean* is *greedily* selected with probability $1 - \epsilon$

- *Thompson Sampling* [50]: according to the original algorithm formulation, each cluster c_k is modelled by means of a Beta distribution $(\alpha_{c_k}, \beta_{c_k}) = (1, 1)$ updated against user feedback according to the following rule:

$$\begin{cases} c^*, & \text{with probability } (1 - \epsilon) \\ c_i \sim \mathcal{C}, & \text{with probability } (\epsilon) \end{cases} \quad (16)$$

$$(\alpha_{c_k}, \beta_{c_k}) += (r_{A_t} 1 - r_{A_t}) \text{ with } A_t \equiv c_k. \quad (17)$$

In each round t, all the Beta distributions are sampled, and the cluster corresponding to the highest value is selected.

- *Upper Confidence Bound (UCB)* [48]: the cluster with the highest upper statistical bound is chosen in each round, adopting an optimistic view which privileges insufficiently explored clusters. UCB algorithms either select the cluster with highest *experimental mean* or highest uncertainty in its estimate, by choosing $A_t = argmax_{c_k} \mu_{c_k} + Bound(c_k, t)$ for each round t. Different algorithms adopt distinct upper

bounds, which have diverse impacts on the cumulative regrets' lower bound of the policy. *UCB1* [48] defines its confidence bound as follows:

$$UCB1(c_k, t) = \sqrt{\frac{\alpha ln(t)}{\eta_{c_k}}}, \qquad (18)$$

where $\alpha > 0$ is an exploration rate. As the number of rounds increases, the confidence bound of the most performed actions decreases exponentially, which means that we are more confident about the estimate of their average reward. In *UCB2* [48], the rounds are divided into epochs. In each new epoch, a single action a_k is performed for a number $\tau(\gamma_i + 1) - \tau(\gamma_i)$ of times, where τ is an exponential function and γ_k is the number of epochs in which the a_k action was performed. The confidence bound is defined as follows:

$$UCB2(c_k, t) = \sqrt{\frac{(1+\alpha)ln(et/\tau(\gamma_i))}{2\tau(\gamma_i)}}, \qquad (19)$$

where $\tau(r) = [(1+\alpha)^r]$. The novelty of *UCB2* as compared to *UCB1* is to ensure that we test the same arm for a significant amount of time before trying a new one, by also allowing to periodically take a break from exploitation, in order to re-explore the other arms.

Another method, *KL-UCB* [51] uses Kullbeck-Leibler divergence to compute the upper confidence bound for the arms, which has been shown to reach a higher cumulative reward for short time horizon T [52]. The *KLUCB* confidence bound is defined as follows:

$$KL(c_k, t) = max\{\widehat{\mu} \in [0,1]$$
$$: kld(\mu_{c_k,t-1}, \widehat{\mu}) \le log_2(t)\} \qquad (20)$$

where $kld(p, q) = p \cdot log(\frac{p}{q}) + (1-p) \cdot log(\frac{1-p}{1-q})$ is the Kullback-Leibler divergence [51].

The last approach of the UCB paradigm considered in this work is the MOSS [53] algorithm. With respect to *UCB1*, this method takes into account the number of previous choices of an arm in formulating its confidence bound instead of the general number of elapsed rounds. This ensures a narrower exploration phase. The confidence bound for the *MOSS* algorithm is defined as follows:

$$MOSS(c_k, t) = \sqrt{\frac{max(log(\frac{T}{t\eta_{c_k}}), 0)}{\eta_{c_k}}} \qquad (21)$$

- Exp3 [54] and *Softmax* [55]: the cluster pulling chance is proportional to an exponential function of either its cumulative reward or experimental mean:

$$\begin{cases} p_{t,c_k} = \dfrac{exp(\lambda\sigma_{t-1,c_k})}{\sum_{j=1}^{K} exp(\lambda\sigma_{t-1,c_j})} & Exp3 \\[3mm] p_{t,c_k} = \dfrac{exp(\lambda\mu_{c_k})}{\sum_{j=1}^{K} exp(\lambda\mu_{c_j})} & Softmax \end{cases}, \qquad (22)$$

where $\lambda > 0$ is the learning rate, and $\sigma_{t,c_k} = \sum_{i=1}^{t} r_{A_i} \ with \ A_i \equiv c_k$ is the cumulative reward for selecting c_k observed until round t.

As will be discussed in Section 4.4, KLUCB achieves the best values for the considered performance metrics; thus, it is selected as the MAB policy underneath the *Recommendation Agent*.

4 | EXPERIMENTAL EVALUATION

In this section, we discuss the experimental setting for evaluating our proposal, describing the adopted datasets (Section 4.1) and performance metrics (Sec.4.2). Then, we perform preliminary tuning of the hyperparameters underneath the MAB policies adopted (Section 4.3), and we evaluate the performances of different MAB policies underneath our system's recommendation agent, by either considering or neglecting the presence of contextual information (Section 4.4). Finally, we compare the performances of our method with different state-of-the-art itinerary recommender systems (Section 4.5).

4.1 | Datasets

In order to assess the effectiveness of our approach, we exploited 5 public datasets[1] of geo-tagged PoIs selected in theme parks (Disneyland, Epcot, California Adventure, Disney Hollywood and Magic Kingdom) [13]. The theme park datasets exhibit a *profit* score for the PoIs which represents their popularity among users. We use this score similarly to the crowdedness information provided by the monitoring infrastructure of our system.

Moreover, we also validate our system against a dataset of mobility traces within a simulated Smart Campus (the University of Palermo, UniPa) [11]. A *smart campus* [56, 57] is a digitally augmented campus where pervasive instrumented objects and spaces are made responsive to the state of the environment and its inhabitants. In a *smart campus*, in addition to the ubiquity of users' smartphones [58, 59], several other sensory devices, such as cameras, RFID readers, bluetooth beacons, and Wireless Sensor Networks collect raw measurements that can be exploited by an intelligent system in order to reason upon current context and supply advanced services to users. For these reasons, a *smart campus* is a great environment to test trajectories' recommendation applications, since there are thousands of students who perform various daily

---

[1] https://sites.google.com/site/limkwanhui/datacode

activities within it, and they would benefit from tailored PoIs' suggestions.

Using synthetic datasets also allows to support the empirical evaluation of the clustering phase. We simulated the mobility traces for 150 users during an academic year, each user having 165 trajectories on average. The dataset contains 54 PoIs belonging to eight different semantic classes present in the dataset. Basically, each user's PoI transition pattern is modelled by means of a different fifth order Markov model for each day of the week. Considering the set of 9 weekly behavioural features described in 3.1, we evaluated different clustering quality metrics: the *adjusted rand index* (ARI) [60] and *adjusted mutual information score* (AMI) [61] which measure the similarity between true labels and predicted labels assignment; the *homogeneity* (HO) [62] which measures whether a cluster contains only members of a single class; the *completeness* (CO) [62] which measures the portion of members of a given class assigned to the same cluster; and finally, the *v-measure* (VM) [62] which is the harmonic mean between the *Completeness* and the *Homogeneity* measures. As already mentioned (see Section 3.1), we adopt the Quality Threshold Clustering (QTC) algorithm, with $m = 4$ (3% of users in the synthetic dataset) and $d = 0.247$ (mean value of the inter-users distances matrix). With these parameters, QTC extracts 9 clusters of users, achieving the quality metrics values reported in Table 3. Similar values for *HO* and *CO* (0.446 and 0.429, respectively) suggest that the clustering is not biased towards errors in mapping every single ground-truth class of users to a unique cluster (*CO*) or viceversa (*HO*). Positive and similar values for *ARI* and *AMI* (0.319 and 0.343, respectively) point out how *QTC* performs well in assigning a cluster to groups of ground-truth users of the same dimension (*ARI*) and unbalanced dimension (*AMI*) [63].

The descriptive statistics of all the adopted datasets are presented in Table 4. All the datasets are preprocessed by filtering out users with less than 2 distinct trajectories in their histories. The profiling phase for the theme park datasets cannot benefit from a weekly computation of the mobility features, since the average number of trajectories per user is included in the range [3, 5]. Therefore, for these datasets, we cluster users considering the feature f_9 as described in Section 3.1 without any weekly embedding. In particular, QTC extracts the following number of clusters for the theme park datasets: $Disneyland \rightarrow 7, Epcot \rightarrow 7, California Adventure \rightarrow 11,$ $Disney Hollywood \rightarrow 10, Magic Kingdom \rightarrow 7$. These results suggest that even with several hundreds of users contributing to the trajectories dataset, the different mobility behaviours can be traced back to a limited range of general habits.

4.2 | Performance metrics

Since we state the trajectory recommendation as a Reinforcement Learning problem, cumulative *reward* and *regret* can be used to assess the algorithm effectiveness by means of the *Damerau-Levenshtein* similarity between recommended and followed trajectories, as discussed in Section 3.2. Moreover, the

TABLE 4 Relevant features of the datasets adopted for evaluation

Features	Datasets					
	(a)	(b)	(c)	(d)	(e)	(f)
Users	150	1006	584	600	470	886
Trajectories	24,824	5458	2275	2870	1663	3738
Unique trajectories	9653	3880	1547	2078	1174	2771
Average trajectories per user	165	5	3	4	3	4
PoIs	54	30	17	25	13	27
PoIs semantic classes	8	6	5	7	4	7
User-trajectory matrix sparsity	0.982	0.998	0.997	0.997	0.997	0.998

Note: Users with less than 2 distinct trajectories are removed from the datasets. The datasets are as follows: (a) UniPa Smart Campus; (b) Disneyland theme park; (c) Epcot theme park; (d) California Adventure theme park; (e) Disney Hollywood theme park; (f) Magic Kingdom theme park.

most common metrics to evaluate trajectory recommendation systems can be used: *precision*, *recall* and *f-score* in the scenario addressed here that can be defined as follows [31].

Let T_{real} and T_{rec} denote the real-life and the recommended trajectory, respectively, and let P_{real} and P_{rec} denote the set of PoIs in trajectory T_{real} and T_{rec}, respectively. The three metrics can be defined as follows:

$$precision = \frac{|P_{real} \cap P_{rec}|}{|P_{rec}|}$$

$$recall = \frac{|P_{real} \cap P_{rec}|}{|P_{real}|} \qquad (23)$$

$$f\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Evaluation is performed for each recommended trajectory against the ξ most similar (according to the Damerau-Levenshtein similarity) trajectories in the current user history H_t, where ξ is a tunable parameter that we set to 5, given that the average number of trajectories per user in five of the six datasets we considered is lower than 5 (see Table 4). A variation of the leave-one-out evaluation is used, where in every iteration, we consider the current user trajectories history as test set, and the histories of all the other users as training set.

4.3 | Recommendation agent parameters tuning

The input parameters of the Multi-Armed Bandit policies adopted in this study vary according to the application scenario. For evaluating the performance of the UCB1 algorithm, the exploration rate has been set as $\alpha = 2$, according to the first proposal of the method [48]. Conversely, in order to select the best parameters' values for the ϵ-greedy, UCB2, Exp3 and Softmax algorithms, we performed a comparative analysis of their *f-score* achieved in different datasets. This evaluation

has been performed by running $T = 100$ trials for each user of the six different datasets, considering also the contextual information. The ϵ parameter in ϵ-greedy represents the probability to explore a new cluster; for this reason, we tested values within a range $[0, 1]$. The α parameter in UCB2 controls the amount of exploration performed by the policy, the higher α the higher the upper confidence bound for a given cluster, thus increasing its probability of being explored; we tested the values within a range $[0.1, 0.9]$, as suggested in the original study [48]. Finally, the learning rate λ in Exp3 and Softmax controls the focus of the policy on the arm with the largest cumulative reward and mean, respectively, so that the higher λ the higher the exploitation phase of the most profitable arm; also in this case, we verified the values in a range $[0.1, 1]$.

The results of these experiments are summarised in Figure 5, which shows the f - $score$ obtained for the ϵ-greedy, UCB2, Exp3 and Softmax policies by varying their input parameters. Even though slightly different values are obtained in the different dataset, a common trend can be noticed. In particular, the best parameter values are those which give more emphasis to the exploitation phase w.r.t. the exploration phase. These are values close to 0.1 for ϵ in ϵ-greedy, and α in UCB2. On the other hand, values of λ close to 0.9 in Exp3 and

Softmax cause a longer exploitation phase which ensures a more effective cluster recognition. The same trend will be confirmed by the results discussed in Section 4.4, since policies with narrower exploration phases allowed to reach the highest values for all the considered performance metrics. The values highlighted in cyan in Figure 5 have been chosen as input parameters to the respective policies during the experiments described in the following subsection.

4.4 | Recommendation agent experimental results

The performances of the recommendation agent were assessed while varying the selected multi-armed bandit policies. Since our system extracts trajectories from clusters according to their frequency of occurrence, we perform 10 distinct experiments with $T = 100$ trials for each user (i.e. 100 trajectories are recommended to each user) in order to mitigate the effects of chance on the evaluation. We consider the (lat, lng) pair of the first PoI in each trajectory as the current position of the current user in order to compute the total distance travelled in a single trajectory.
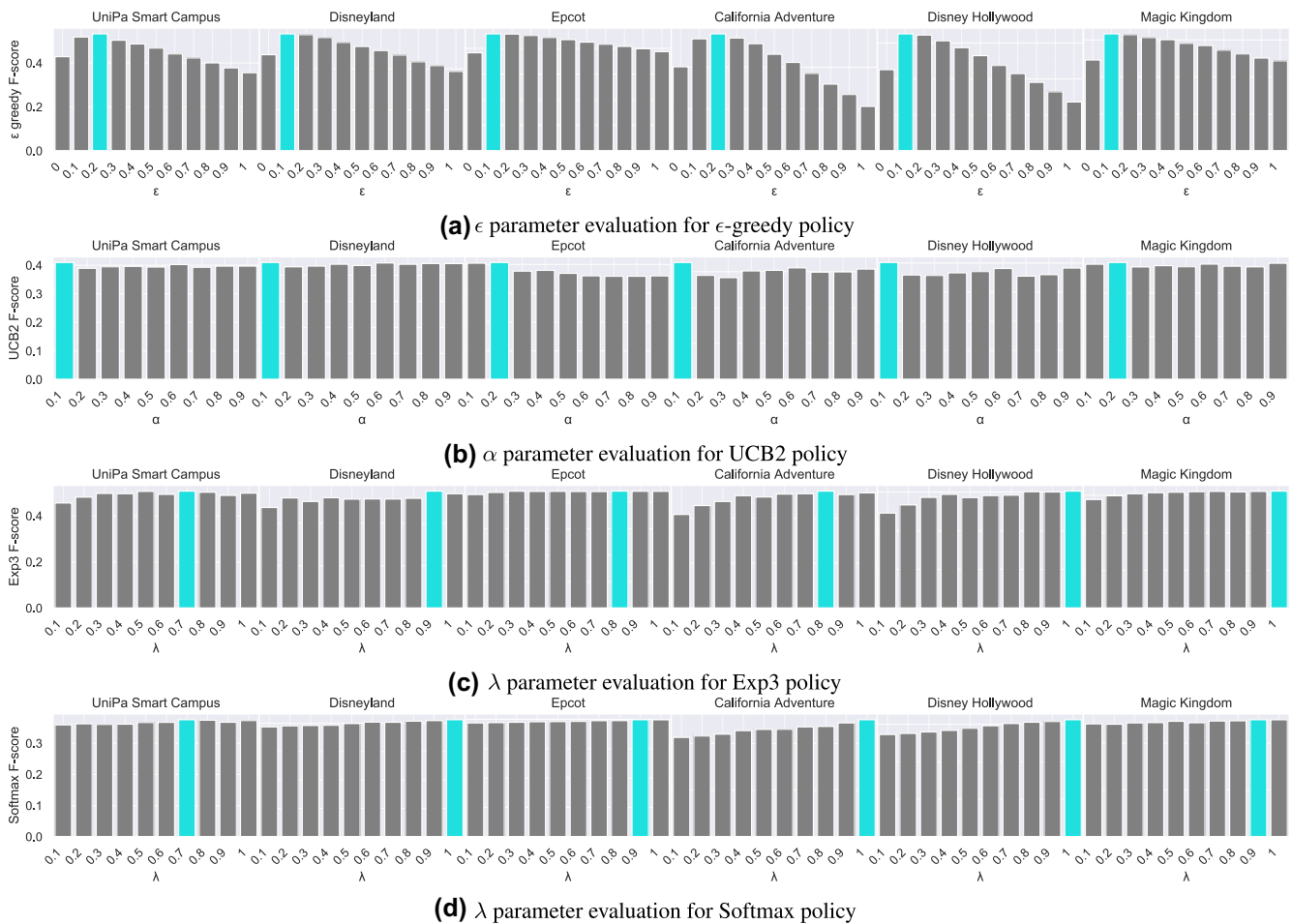

(a) ϵ parameter evaluation for ϵ-greedy policy

(b) α parameter evaluation for UCB2 policy

(c) λ parameter evaluation for Exp3 policy

(d) λ parameter evaluation for Softmax policy

FIGURE 5 Multi-Armed Bandit policies parameters tuning against the six adopted datasets. The bars highlighted in cyan, represent the value chosen for the corresponding policy

Generally, considering user context actually improves the cumulative *reward*, *precision*, *recall* and *f-score* for every MAB policy, while also decreasing the cumulative *regret*. In particular, averaging the mean values across all datasets and MAB policies, taking into account contextual information allows to achieve mean *reward*, *precision*, *recall* and *f-score* score values of about 51%, 50%, 32%, 52%, respectively, while achieving a lower *regret* mean value of about 35%; this suggests that context information is important in determining the correct users cluster membership. For our UniPa Smart Campus dataset, the improvement in considering contextual information reflects our simulator implementation [11] where users typically prefer to reach closer PoIs under the same semantic class, as suggested in a related research study [49]. The cumulative *reward* results exhibit a clear lower pattern of values with respect to *precision* against the theme park datasets, this is due to the nature of the *Damerau-Levenshtein* similarity measure employed in our computation of the agents rewards, as it also takes into account the temporal arrangement of the PoIs, penalising the asynchronous sequences between recommended and actual trajectories. This aspect is neglected by

precision and *recall* measures, and the reason why this temporal aspect does not affect our synthetic dataset *reward* values is that the high amount of trajectories per users has given us the opportunity to cluster users with a more fine grained resolution, taking into account the temporal aspects of their weekly movement behaviour patterns, as we discussed in Section 4.1.

In regard to the assessment of the performances of the MAB policies against the various datasets, Figure 6 summarises cumulative *reward* and *regret*, *precision*, *recall* and *f-score* values. Each coloured bar represents the mean value of the corresponding metric, with error bars which show the 95% confidence interval of the mean. The random policy uniformly pulls every cluster without any heuristics, thus collecting the lowest performance metrics. The ϵ greedy policy is the second best performing policy, pointing out how even with few iterations, the agent is able to assign high probability to the real cluster of the current user. Thompson Sampling have a wider exploration stage with respect to ϵ greedy, thus collecting a slightly lower accuracy. Amid UCB algorithms, KLUCB has the highest

(a) Cumulative *reward* and *regret*

(b) *precision*, *recall* and *f-score*

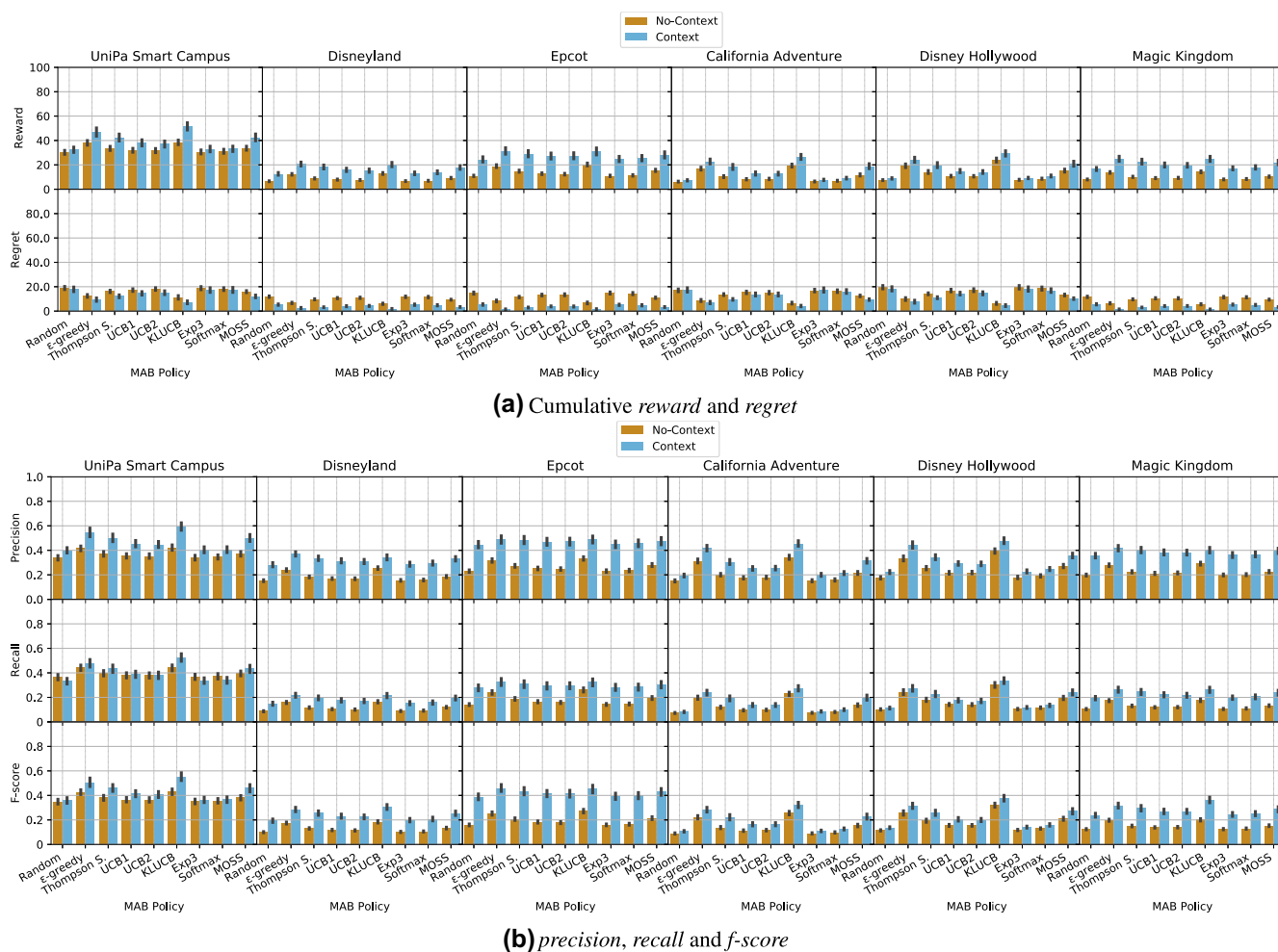**FIGURE 6** Performance metrics of different MAB policies against the adopted datasets. The MAB policies are as follows: (1) Random; (2) ϵ- greedy; (3) Thompson Sampling; (4) UCB1; (5) UCB2; (6) KLUCB; (7) Exp3; (8) Softmax; and (9) MOSS

performance metrics values, and this is attributable to the Kullback-Leibler divergence which significantly increases the upper confidence bound value for the highest rewarding cluster, thus narrowing the exploration phase of incorrect clusters. With regard to policies with exponential pulling probabilities, Exp3 exhibits higher metrics values with respect to Softmax, suggesting that focussing on the arms' experimental means allows for a wider exploitation of the real users' clusters. Summarising, best performances for the trajectory recommendation problem with MAB policies are achieved with KLUCB, and in general, with policies having narrower exploration phases. This highlights how user feedback has a high informative value on its cluster membership and the potentially rewarding trajectories. In light of these results, we consider KLUCB as the MAB policy beneath our *Recommendation Agent*.

4.5 | Comparative analysis

In this section, we compare our approach against three state-of-the-art itinerary recommendation baselines. In order to perform a fair comparison, we follow the experimental protocol described in [13] considering, for each trajectory in the ground truth, a recommended trajectory that shares the starting and ending PoIs. For this simulation, we adopt the KLUCB policy for the recommendation agent as it resulted to be the most proficient policy (see Section 4.4), we then consider the travelled distance as contextual information for the monitoring agent. As baselines, we selected the following three methods:

- *PersQ* [13]: uses the Monte Carlo Tree Search method to find an itinerary maximising the interest of the user (calculated considering the labels of the PoIs in the training itineraries), the popularity of the visited PoIs (calculated as the frequency of visit for all users in the dataset), as well as minimising the queueing time at the PoIs (calculated starting from the knowledge of how long it takes to make a turn in the particular attraction);
- *IHA* [64]: employs a heuristic search algorithm to propose the itinerary which maximises both user interests and PoIs popularity, respecting a given user's budget in terms of travelling duration;
- *PersTour* [65]: infers user interests from its visit durations with a comparison to other users, it then employs two weighting approaches to evaluate the contribution of PoIs popularity and interest preferences for the active user.

In order to make a clear comparison with the state of the art, experiments were performed by exploiting the 5 public datasets discussed so far. Table 5 shows the *precision*, *recall* and *f-score* of *PersQ* [13], *IHA* [64], *PersTour* [65], against our approach. The method we propose exhibits the highest *precision* values for all the considered datasets, with an improvement which ranges from 39.2% to 181.6% with respect to the baselines. Considering the *recall*, our method

TABLE 5 Comparisons in terms of precision, recall and f-score with state-of-art baselines against 5 real life theme park datasets [13]

Dataset/Metrics	Our approach			PersQ [13]			IHA [64]			PersTour [65]		
	Precision	Recall	f-score	Precision	Recall	f-score	Precision	Recall	f-score	Precision	Recall	f-score
Disneyland	**0.478 ± 0.006**	**0.412 ± 0.005**	**0.319 ± 0.005**	0.295 ± 0.006	0.332 ± 0.006	0.289 ± 0.005	0.270 ± 0.004	0.267 ± 0.004	0.249 ± 0.003	0.201 ± 0.005	0.190 ± 0.004	0.180 ± 0.004
Epcot	**0.575 ± 0.008**	**0.478 ± 0.008**	**0.412 ± 0.007**	0.413 ± 0.012	0.472 ± 0.012	0.407 ± 0.010	0.368 ± 0.008	0.380 ± 0.008	0.353 ± 0.007	0.322 ± 0.009	0.310 ± 0.008	0.297 ± 0.007
California adventure	**0.539 ± 0.008**	0.409 ± 0.007	**0.365 ± 0.007**	0.307 ± 0.007	**0.483 ± 0.010**	0.338 ± 0.007	0.296 ± 0.006	0.332 ± 0.006	0.287 ± 0.005	0.223 ± 0.007	0.227 ± 0.007	0.204 ± 0.006
Disney Hollywood	**0.615 ± 0.010**	0.444 ± 0.009	0.403 ± 0.008	0.439 ± 0.014	**0.482 ± 0.015**	**0.431 ± 0.013**	0.417 ± 0.012	0.367 ± 0.010	0.371 ± 0.010	0.367 ± 0.012	0.305 ± 0.009	0.317 ± 0.010
Magic Kingdom	**0.566 ± 0.007**	0.399 ± 0.006	**0.376 ± 0.006**	0.326 ± 0.007	**0.440 ± 0.008**	0.343 ± 0.006	0.312 ± 0.006	0.305 ± 0.005	0.288 ± 0.004	0.201 ± 0.006	0.202 ± 0.005	0.186 ± 0.005

shows an improvement ranging from 1.3% to 116.8% compared to the other baselines, with the exceptions for the California Adventure, Disney Hollywood and Magic Kingdom datasets, where our approach achieved lower *recall* values compared to *PersQ*. However, our approach outperforms *PersQ*'s in terms of *fscore* in all the dataset, except Disney Hollywood. The different results provided by the two approaches can be explained by analysing the general recommendation paradigm they adopt. In particular, we chose a *collaborative filtering* paradigm, meaning that only trajectories already travelled by other users in the dataset will be recommended to the current user. Conversely, *PersQ* belongs to the *content-based* paradigm, meaning that the approach may propose trajectories never travelled before and which may be of potential interest for the current user. These considerations both implies that in our approach, the suggested trajectories will not overfit on the particular current user's interests (causing lower *recall* values), and that they will instead be effectively travelled trajectories coming from the identified cluster of similar users (causing higher *precision* values). Finally, our approach has greater versatility than other methods, which consists in letting the user cluster membership change over time based on feedback, and in overcoming the need for providing the starting and ending PoIs of the itinerary, which are instead required in *PersQ* and *IHA*.

5 | CONCLUSIONS

In this study, we presented a novel approach for trajectory recommendation in smart environments. Suggestions are provided by exploiting two agents that are responsible for the profiling phase, where users are grouped on the basis of similar behaviours, and for the recommendation phase, where the associations between groups and itineraries are computed by solving a Multi-Armed Bandit problem. Differently from other works that are mainly based on geographic features, the proposed system also exploits context information in the form of travelling distance and crowdedness of PoIs. Moreover, one major contribution lies in the adoption of the reinforcement learning paradigm for modelling the trajectory recommendation problem as a MAB policy, which allows to reduce the amount of information required from users in order to drive the collaborative filtering process. We assessed the effectiveness of our approach against six datasets, measuring five different metrics, and comparing the results with those achieved by three different state-of-the-art itinerary recommendation methods. The experimental results showed the importance of context information in providing more relevant recommendations, and the superior accuracy of MAB policies with narrower exploration phases, for example KLUCB. The choice of the proper context and its integration with the MAB policy represent the most critical points for guaranteeing the effectiveness of the suggestions. Future work can concern the evaluation of different sources of contextual information. For instance, assuming the system is aimed to provide suggestions to the students of a smart campus, relevant information can be inferred by analysing the lessons timetable, or considering the activities performed by friends or colleagues. Moreover, context information can be exploited directly from the MAB policies beneath our recommendation agent, using heuristics such as LinUCB [66] or Exp4 [54].

ORCID

Marco Morana https://orcid.org/0000-0002-5963-6236

REFERENCES

1. Casella, E., et al.: Hierarchical Syntactic Models for Human Activity Recognition Through Mobility Traces, pp. 1–14. *Personal and Ubiquitous Computing* (2019)
2. Aggarwal, C.C.: Recommender systems - The textbook, 1. Springer (2016)
3. Gaglio, S., et al.: A fast and interactive approach to application development on wireless sensor and actuator networks. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation, pp. 1–8. ETFA (2014)
4. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: technologies, applications, and opportunities. Pervasive Mob Comput. 5(4), 277–298 (2009)
5. Agate, V., et al.: Smart services for an augmented campus. In 2018. IEEE International Conference on Smart Computing (SMARTCOMP), 276–278 (2018)
6. Chu, W., Park, S.: Personalized recommendation on dynamic content using predictive bilinear models. In Proceedings of the 18th international Conference on World wide web, pp. 691–700. (2009)
7. Wang, Q., et al.: Online interactive collaborative filtering using multi-armed bandit with dependent arms. IEEE Trans. Knowl. Data. Eng. 31(8), 1569–1580 (2018)
8. Zhang, R., et al.: Differentiating population spatial behaviour using representative features of geospatial mobility (refgem). ACM Trans. Spatial Algorithms Syst. 6(Feb) (2020)
9. Zhong, S., et al.: An adaptive similarity-measuring-based cmab model for recommendation system. IEEE Access. 8, 42550–42561 (2020)
10. Li, L., et al.: A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web, pp. 661–670. (2010)
11. De Paola, A., et al.: Human mobility simulator for smart applications, In The 23rd IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications. DS-RT, pp. 1–8. (2019)
12. De Paola, A., et al.: Detection of points of interest in a smart campus, In 5th international Forum on Research and Technologies for Society and Industry: Innovation to Shape the future, pp. 155–160. (2019)
13. Lim, K.H., et al.: Personalized itinerary recommendation with queuing time awareness. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR, 17, pp. 325–334. (2017)
14. Joseph, K., Tan, C.H., Carley, K.M.: Beyond local, categories and friends: clustering foursquare users with latent topics. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 919–926. UbiComp 12 (2012)

15. Ghosh, J., et al.: On profiling mobility and predicting locations of wireless users. In Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, pp. 55–62. (2006)

16. Dang, T.A., Chiam, J., Li, Y.: A comparative study of urban mobility patterns using large-scale spatio-temporal data. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 572–579 (2018)

17. Rodriguez-Carrion, A., et al.: Impact of location history collection schemes on observed human mobility features. IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS), pp. 254–259. IEEE (2014)

18. Thuillier, E., et al.: Clustering weekly patterns of human mobility through mobile phone data. IEEE Trans. Mobile Comput. 17, 817–830 (2018)

19. Yin, H., et al.: Lcars: a location-content-aware recommender system. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 221–229. (2013)

20. Song, C., et al.: Limits of predictability in human mobility. Science. 327, 1018–21 (2010)

21. Yuan, Q., et al.: Time-aware point-of-interest recommendation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 363–372. SIGIR 13 (2013)

22. Gavalas, D., et al.: A survey on mobile tourism recommender systems, In 2013 Third International Conference on Communications and Information Technology (ICCIT), pp. 131–135. IEEE (2013)

23. Gavalas, D., et al.: A survey on algorithmic approaches for solving tourist trip design problems. J. Heuristics. 20(3), 291–328 (2014)

24. Lim, K.H., et al.: Tour recommendation and trip planning using location-based social media: a survey. Knowl. Inf. Syst. 60(3), 1247–1275 (2019)

25. Yin, H., et al.: Joint modeling of user check-in behaviours for point-of-interest recommendation. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1631–1640. CIKM 15 (2015)

26. Yao, L., et al.: Context-aware point-of-interest recommendation using tensor factorization with social regularization. In Proceedings of the 38th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1007–1010. SIGIR 15 (2015)

27. Cheng, C., et al.: Where you like to go next: successive point-of-interest recommendation. In Twenty-Third International Joint Conference on Artificial Intelligence (2013)

28. Gionis, A., et al.: Customized tour recommendations in urban areas. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 313–322. WSDM 14 (2014)

29. Fogli, A., Sansonetti, G.: Exploiting semantics for context-aware itinerary recommendation. Pers. Ubiquitous Comput. 23, 215–231 (2019)

30. Zhao, P., et al.: Photo2trip: exploiting visual contents in geo-tagged photos for personalized tour recommendation. IEEE Trans. Knowl. Data Eng. 33(4), 1708–1721 (2019)

31. Chen, L., et al.: Personalized itinerary recommendation: deep and collaborative learning with textual information. Expert Syst. Appl. 144, 113070 (2020)

32. Lim, K.H., et al.: Towards next generation touring: personalized group tours. In Proceedings of the International Conference on Automated Planning and Scheduling. 26(n. 1) (2016)

33. Anagnostopoulos, A., et al.: Tour recommendation for groups. Data Min. Knowl. Discov. 31(5), 1157–1188 (2017)

34. Hu, F., et al.: Agree: attention-based tour group recommendation with multi-modal data. In International Conference on Database Systems for Advanced Applications, pp. 314–318. Springer (2019)

35. Bifulco, I., Cirillo, S.: Discovery multiple data structures in big data through global optimization and clustering methods. In 2018 22nd International Conference Information Visualisation (IV), pp. 117–121. (2018)

36. Wasid, M., Ali, R.: Multi-criteria clustering-based recommendation using mahalanobis distance. Int. J. Reas. base Intell. Syst. 12(2), 96–105 (2020)

37. Banda, L., Singh, K.: Improving scalability, sparsity and cold start user issues in collaborative tagging with incremental clustering and trust. Recent Advances in Computer Science and Communications (Formerly: Recent Pat Comput Sci. 14(2), 429–438 (2021)

38. Sanz-Cruzado, J., Castells, P., López, E.: A simple multi-armed nearest-neighbor bandit for interactive recommendation. In Proceedings of the 13th ACM Conference on Recommender Systems, pp. 358–362. (2019)

39. Martorella, G., Peri, D., Toscano, E.: Hardware and Software Platforms for Distributed Computing on Resource Constrained Devices, pp. 121–133. Springer International Publishing, Cham (2014)

40. Peri, D.: Body Area Networks and Healthcare, pp. 301–310. Springer International Publishing, Cham (2014)

41. Agate, V., Ferraro, P., Gaglio, S.: A cognitive architecture for ambient intelligence systems. In CEUR Workshop Proceedings. 2418, 52–58 (2019)

42. Restuccia, F., et al.: A framework for optimizing information quality in mobile crowdsensing systems. ACM Trans. Sens. Netw. 15(1) (2018)

43. Ferraro, P., Lo Re, G.: Designing Ontology-Driven Recommender Systems for Tourism, pp. 339–352. Springer International Publishing, Cham (2014)

44. Alvares, L.O., et al.: Towards semantic trajectory knowledge discovery. Data Min. Knowl. Discov. 12 (2007)

45. Gonzalez, M.C., Hidalgo, C., Barabasi, A.L.: Understanding individual human mobility patterns. Nature. 453, 779–82 (2008)

46. Bard, G.V.: Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In Proceedings of the Fifth Australasian Symposium on ACSW Frontiers. 68, pp. 117–124. ACSW 07 (2007)

47. Heyer, L.J., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome Res. 9(11), 1106–1115 (1999)

48. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multi-armed bandit problem. Mach. Learn. 47, 235–256 (2002)

49. Lee, K., et al.: Slaw: a new mobility model for human walks. IEEE INFOCOM, 855–863 (2009)

50. Chapelle, O., Li, L.: An empirical evaluation of thompson sampling. In Advances in Neural Information Processing Systems. 24, pp. 2249–2257. Curran Associates, Inc. (2011)

51. Garivier, A., Cappé, O.: The kl-ucb algorithm for bounded stochastic bandits and beyond. In Proceedings of the 24th Annual Conference on Learning Theory, pp. 359–376. (2011)

52. Tanczos, E., Nowak, R., Mankoff, B., A kl-ucb algorithm for large-scale crowdsourcing, In Advances in Neural Information Processing Systems 30 (I. Guyon, et al., eds.), pp. 5894–5903, Curran Associates, Inc., (2017)

53. Audibert, J.-Y., Bubeck, S.: Regret bounds and minimax policies under partial monitoring. J. Mach. Learn Res. 11(94), 2785–2836 (2010)

54. Auer, P., et al.: The nonstochastic multiarmed bandit problem. SIAM J. Comput. 32(1), 48–77 (2002)

55. Cesa-Bianchi, N., et al.: Boltzmann exploration done right. In Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 6287–6296. Curran Associates Inc., NY (2017)

56. Concone, F., et al.: Towards a smart campus through participatory sensing. In 4th IEEE International Workshop on Sensors and Smart Cities. SSC (2018)

57. Gaglio, S., et al.: Smart assistance for students and people living in a campus. In 2019. IEEE International Conference on Smart Computing (SMARTCOMP), 132–137 (2019)

58. Concone, F., et al.: Smartphone data analysis for human activity recognition. In AI*IA 2017 advances in Artificial intelligence, pp. 58–71. Springer International Publishing, Cham (2017)

59. Concone, F., Lo Re, G., Morana, M.: A fog-based application for human activity recognition using personal smart devices. ACM Trans. Internet Technol. 19(2) (2019)

60. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. 2, 193–218 (1985)

61. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn Res. 11, 2837–2854 (2010)

62. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 joint Conference on empirical Methods in natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 410–420. Association for Computational Linguistics, Prague (June 2007)

63. Romano, S., et al.: Adjusting for chance clustering comparison measures. J. Mach. Learn Res. 17(1), 4635–4666 (2016)

64. Zhang, C., Liang, H., Wang, K.: Trip recommendation meets real-world constraints: poi availability, diversity, and traveling time uncertainty. ACM Trans. Inf. Syst. 35(1), 1–28 (2016)

65. Lim, K.H., et al.: Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl. Inf. Syst. 54(2), 375–406 (2018)

66. Chu, W., et al.: Contextual bandits with linear payoff functions. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 208–214. (2011)