

## On the Composition and Recommendation of Multi-Feature Paths: a Comprehensive Approach

Vincenzo Cutrona · Federico Bianchi ·  
Michele Ciavotta · Andrea Maurino

Received: date / Accepted: date

**Abstract** Trackers have become popular devices these days. They are extensively used to record sports activities (e.g., hiking, skiing), mainly in terms of GPS trajectories, which can be shared on social networking platforms with other users looking for leisure tips. Notably, as the number of available trajectories drastically increased over time, in many cases, it has become challenging, if not impossible, the extensive evaluation of all possible alternatives and the manual selection of the most suitable one. Paths are characterized by multiple features (e.g., dirt, asphalt), and a good representation is needed to satisfy user needs. Moreover, paths can be composed to generate new routes. This calls for a recommender system capable to handle both the multi-feature path representation and the implicit definition of alternatives by composition. This paper suggests a novel approach that features a richer trajectory represen-

---

This is a post-peer-review, pre-copyedit version of an article published in GeoInformatica. The final authenticated version is available online at: <https://doi.org/10.1007/s10707-019-00356-z>.

---

Vincenzo Cutrona  
University of Milano-Bicocca  
Viale Sarca, 336  
E-mail: [vincenzo.cutrona@disco.unimib.it](mailto:vincenzo.cutrona@disco.unimib.it)  
ORCID ID: 0000-0001-7830-7596

Federico Bianchi  
University of Milano-Bicocca  
Viale Sarca, 336  
E-mail: [federico.bianchi@disco.unimib.it](mailto:federico.bianchi@disco.unimib.it)

Michele Ciavotta  
University of Milano-Bicocca  
Viale Sarca, 336  
E-mail: [michele.ciavotta@disco.unimib.it](mailto:michele.ciavotta@disco.unimib.it)

Andrea Maurino  
University of Milano-Bicocca  
Viale Sarca, 336  
E-mail: [andrea.maurino@disco.unimib.it](mailto:andrea.maurino@disco.unimib.it)

tation based on a semantic annotation to describe significant path features. Annotations are then used for automatic recommendation of new paths that maximize the presence of characteristics matching the user preferences. Finally, a class of algorithm variants is evaluated using an off-line validation process and compared with a baseline solution to test the underlying assumptions.

**Keywords** recommender systems · multi-feature paths · path composition · gps trajectories

## 1 Introduction

The widespread of low-cost tracking devices (smartphones and smartwatches are equipped with an onboard GPS sensor) has led to the large availability of user-generated geospatial content, among which GPS trajectory is undoubtedly a class that plays a pivotal role. Outdoor enthusiasts have begun to form large and ever-expanding online communities such as *Wikiloc*<sup>1</sup> and *GpsX-Change.com*<sup>2</sup>, whose aim is to foster sharing of suggestions and experiences. For this reason, the number of GPS trajectories freely available is very high (more than 8 million trajectories are available on *Wikiloc* only) and grows at a staggering pace. In this sea of information, however, it has become difficult to find one's way around as users can usually search, filter and select trajectories based on distance and geolocation only. Incidentally, it is a daunting task to create a recommendation system centered in the interests of the single user when paths often have no semantic connotations whatsoever. Additionally, available trajectories are often no more than collections of geolocated points that cannot be used to infer new paths.

In order to clarify the dynamics and the criticalities that a path composition and recommendation system has to cope with, we rely on the following motivating scenario: Chiara is an amateur hiker interested in nature photography. Because of her inexperience, she prefers walking on entry-level difficulty paths, such as dirt (or gravel) roads. She decides to start a new activity in a place that she has not explored before looking for a path tailored to her interests: she is looking for a path that is mostly on a dirt road, that crosses a woodland and avoids technical sections like rock surfaces. Thus, Chiara's problem is to find a path that reflects her interests and that starts in the surrounding area, with possible constraints about the destination and the path length.

Traditional route planning services are incapable of addressing such a problem, due to some limitations: (i) they are designed for searching paths between two points (start and arrival points); (ii) they do not consider users' interests; (iii) these systems rely on pre-modeled route graphs, which represent cities roads with meta-data (e.g., road type, max allowed speed, one-way or two-way road) that do not apply in case of open-air paths (e.g., trails, park footpaths,

---

<sup>1</sup> <https://www.wikiloc.com/>

<sup>2</sup> <http://www.gpsxchange.com/>

off-road paths, off-piste ski). Differently, some recommender systems might suggest points of interest along the path [28,15,4], such as woodlands, still they ignore the paths that connect those points charging the user with the task of identifying the best route. Nevertheless, some recommender systems that suggest trajectories exist [13,21], but they only consider trajectories as a whole. As a result, a path can be recommended only if someone has already walked it, without checking if the same path is derivable by composing several other (sub)-trajectories. Alternatively, Chiara can search for pre-packaged trendy itineraries (which are, for example, suggested by specialized magazines) or she can resort to asking an expert about itineraries that best fit her needs. These solutions (especially the latter) are not always viable or affordable.

From the above analysis, it should be evident that an automatic system for scouting interesting trajectories and recommending them to the right user is desirable. Moreover, such a solution should also be able to suggest routes considering the compositional nature of trajectories, which are, in fact, modular items that can be composed and disaggregated in new paths where the intersection points between trajectories play a relevant role.

While entrenched models to represent roadmaps are available, this is not the case for the open-space domain, where the path underlying the trajectories has to be identified and refined, although recent studies have investigated models for constructing map representations starting from GPS trajectories [11,10,6,1,14,7]. Besides, a dedicated line of research exploited methods for semantically enriching trajectories in order to produce better recommendations. Some methods analyze the trajectory itself and produce a set of labels that better describe the content of a trajectory from a semantic point of view (e.g., by identifying *stop* and *move* points) [28,25]; other approaches prefer to involve users in the annotation task when some labels are not directly deducible from trajectories points [24,23]. Finally, recent works aimed at enriching trajectories by fetching information from external data sources in order to reduce the human effort required in the annotation process [26,20].

In this paper, a comprehensive approach is proposed to i) automatically compose novel routes and ii) recommend them to users. Our solution takes a collection of raw trajectories as input, which are labeled with semantic tags (e.g., rock, dirt); trajectories are combined in a unified graph that is explored to provide users with paths that best satisfy their needs. More in details, an end-to-end paths recommender system has been designed to investigate new models to maximize the user's degree of satisfaction with the recommended routes. The system integrates the multiple steps required to support the process from raw data to recommendation into a single pipeline. Specifically, five principal activities are featured: (i) collection of raw trajectories, either directly from sensors or from data sources represented in standard format<sup>3</sup>; (ii) annotation of trajectories with semantic labels, which are collected from users; (iii) generation of a graph, starting from the annotated paths set, to represent the map underlying the raw trajectories, in such a way as to preserve users an-

---

<sup>3</sup> In this work the well-known GPS eXchange Format (GPX) has been adopted.

notations; (iv) exploration of the graph to retrieve the set of candidate paths, given a geographical starting point (i.e., a latitude-longitude pair). Since in the considered scenarios the arrival point is not always required as the user might be interested in the overall path length rather than its destination, a heuristic approach to estimate the best suitable path length is proposed; (v) estimation of the user rating for each candidate path, based on the user profile, in order to provide the list of the fittest paths. Different models have been studied to compute the estimated user's appreciation for a path; such models analyze the order in which features appear along the path, including the assumption that interesting features at the end of a path are more or less desirable than the same features at the beginning of the path.

Since no gold standards are available in literature due to relative newness of the field, the proposed approach has been evaluated through an experimental setup involving a reference group of 38 real users active in online social network communities, who have been profiled to identify their specific interests and asked to rate the outcome of 7 variations of our approach and a baseline applied over a consolidated map covering an area of 20 km radius in a mountainous region of northern Italy. To assess the quality of the recommendations concerning the users' preferences, the Normalized Discounted Cumulative Gain (nDCG) [12] metric has been exploited, considering the top-5 and top-10 suggested paths. The results demonstrate the soundness of the proposed approach, validating the underlying hypotheses of this study. Notably, results show that a multi-feature representations for paths can be profitably used in recommendation settings; in all the cases examined, in fact, the users show to favour the personalized paths over the baseline, preferring, in addition, a recommendation model that favours the maximization of the global (path-wise) similarity over that of the individual segments. Finally, the experimental analysis also shows, as was to be expected, that more experienced users tend to evaluate the recommended routes more positively than the average, justifying the need for personalized recommendations.

The paper is organized as follows: Section 2 describes the state-of-the-art models of the field. In Section 3 preliminary definitions that will be used throughout the paper are presented. Section 4 introduces the approach, discussing, in particular, the path representation and the path recommendation. In Section 5 the main results of the off-line experiment conducted with real users are presented and discussed. The conclusions along with some possible future work directions are reported in Section 6.

## 2 Related Work

In this section, pertinent literature is analyzed in an attempt to highlight similarities and differences with our research. Since the proposed model is a unified and complete approach for map inference and path recommendation, the presented literature is classified according to three principal topics. Noticeably, the analysis of the literature relating to traditional route planning solutions

has deliberately omitted, as those only cope with quantitative features (e.g., distance, time), without considering the user’s preferences and thus are not suitable for personalized path recommendation.

*Path recommendation.* Recommending personalized paths is a still fresh research area. Most of all works are related to the tourism domain, where the objective is to recommend a list of interesting attractions and places to visit. In [28] the objective is to mine the top- $n$  locations of interest and the top- $m$  classical travel sequences between locations from a set of GPS trajectories; this information is exploited for recommending itineraries to tourists. In [4], the GPS trajectories are used together with the user’s current position to suggest touristic spots. A different approach focuses on the user’s visit duration at sightseeing locations, assuming that if a tourist is interested in a location, the visit duration may be longer [15]. Other researches propose models to mine pattern underlying users’ travels. A study about the extraction of semantic trajectories from a sequence of geotagged photos is discussed in [5], where the authors propose a Collaborative Filtering (CF) based recommender system. A similar approach is introduced in [9], where the trajectories are analyzed for estimating users’ travel behavior frequencies, which are then used to suggest routes via a Bayes model. Other works in literature also consider means of transportation, suggesting multi-modal routes based on users’ preferences [13, 21]. It is important to notice that most of all works mentioned above aim at suggesting a list of points of interest, while the objective of this work is to recommend paths featuring interesting traits for the users.

*Semantic trajectories enrichment.* The scarcity of features for paths characterization has prompted research activities towards investigating alternative representations of trajectory. Many studies proposed a semantic label-based representation of trajectories, i.e., to annotate a trajectory with a set of labels that provides aspatial semantic information. The labeling process is performed automatically in some cases [28, 25], by analyzing the trajectories points and mining aspatial features (e.g., a set of points in a trajectory can be labeled as *stop* if the recorded object stayed in the same area for a while, or as *mean of transportation* if the speed of the moving object exceeds a specified threshold). Other works are focused on features that are not directly identifiable with raw trajectory analysis (e.g., environmental features). For this reason, these approaches require human users input in order to annotate trajectories properly [24, 23]. In the last few years, many works started enriching trajectories by retrieving information from external data sources, such as environmental features to suggest healthy paths [21], visual information extracted from images available in Google Street View to improve the diversity of the path [26], or entities described in a Knowledge Base (KB) to enrich the Points of Interest found along the trajectory [20].

*Map inference and trajectory clustering.* As large collections of GPS trajectories become available, building and updating maps out of such sources grew

in popularity within the academia. While some approaches are focused on detecting common sub-paths [16,3,17], other works addressed the problem of reconstructing the map underlying the trajectories. This task is addressed mainly following three different approaches: point-clustering [11,10], incremental track insertion [6,1] and insertion linking [14]. A recent study presented a supervised learning framework to create up-to-date maps with rich knowledge from large GPS trajectory collections, which efficiently constructs high-quality maps at a city scale [7].

To conclude the section, some considerations regarding the total lack of scientific approaches to the problem using Deep Learning (DL) models are discussed. Interestingly, even if such approaches could also be used in this context, to the best of our knowledge, none of the existing path recommender systems exploit them. The main reason is that DL approaches would suffer from some limitations, briefly summarized in the following. DL models often require large amounts of data to make inferences, while approaches like ours do not require a training phase and can be easily integrated into recommendation contexts. Also, DL approaches have difficulties in dealing with a user changing opinions (i.e., in our system if a user decides that she prefers *dirt* with respect to other elements, after weeks of using the systems, she can update her profile, and no further training is needed). One last minor consideration on DL models is related to the explainability of the suggestions: the output of DL models comes from different matrix operations and explaining outputs is still an open research problem for the DL community. Our model deals with an explicit representation of route states and user preferences, and thus its suggestions are more explainable.

### 3 Definitions and assumptions

Most of all tracking devices record itineraries as (raw) *trajectories*, which can be formally defined as follows:

**Definition 1** A **trajectory**  $T = \langle p_1, p_2, \dots, p_n \rangle$  is an ordered list of points  $p_i = \langle x_i, y_i, t_i \rangle$ , where  $x_i, y_i$  represent the latitude and the longitude of the  $i^{th}$  point, and  $t_i$  is its acquisition timestamp.

Since sometimes users record more than one activity as one long session, i.e., the tracker is paused when the user is stationary (this can happen when the user stops near to a point of interest, like a shop or a restaurant), the raw trajectory requires preprocessing to identify the real path behind the data points. Thus, starting from a trajectory and given a time window  $\delta_t$ , a partition of sub-trajectories (named *paths*) can be identified, each of which can be defined as follows:

**Definition 2** A **path**  $P_{\delta_t} = \langle p_1, p_2, \dots, p_n \rangle$ , where  $p_i \in T$ , is a subsequence of a trajectory  $T$ , where the following condition holds:  $\forall p_i, p_{i+1} \in T, p_{i+1}.t - p_i.t < \delta_t$ , where  $p_i.t$  represents the timestamp associated with point  $p_i$ .

Also, a path can be divided again into sub-paths (named *segments*), based on some arbitrary conditions.

**Definition 3** A **segment**  $S = \langle p_1, p_2, \dots, p_n \rangle$  is a subsequence of consecutive points in a path  $P$ .

**Definition 4** Given a distance function  $dist(p_i, p_j)$  and a threshold  $\delta_s$ , an ordered list of segments  $X = \langle S_1, S_2, \dots, S_n \rangle$  represents a **valid path** if the following conditions hold:

$$\forall S_i, S_{i+1} \in X, dist(S_i.p_n, S_{i+1}.p_1) < \delta_s \quad (1)$$

$$\forall S_i, S_{i+1} \in X, S_i \cap S_{i+1} = \emptyset \quad (2)$$

where  $S_i.p_n$  represents the last point of segment  $S_i$  and  $S_{i+1}.p_1$  is the first point of segment  $S_{i+1}$ .

Let us assume that the user is provided with the tools to select an arbitrary segment of the trajectory and annotate it with one or more labels, referring to a specific path trait, in order to reveal features not directly measurable by the device (e.g., the user could label a segment using the *dirt* and *sun* tags to describe that segment as *a dirt path that is also sun exposed*). Given a predefined set of labels  $\mathcal{L}$ , a labeled segment can be defined as follows:

**Definition 5** A **labeled segment**  $LS$  is a pair  $LS = \langle S, L \rangle$ , where  $S$  is a segment and  $L = \{l_1, l_2, \dots, l_m\} \subseteq \mathcal{L}$  is the set of labels for all points  $p_i \in S$ .

Starting from the previous definition, the concept of the labeled path can be defined accordingly.

**Definition 6** A **labeled path**  $LP = \langle LS_1, LS_2, \dots, LS_n \rangle$  is an ordered list of labeled segments where conditions 1 and 2 hold,  $\forall LS_i.S, LS_{i+1}.S \in LP$ .

It is important to remark that the labels set of an annotated segment  $L \subseteq \mathcal{L}$  can be seen as a binary vector of size  $|\mathcal{L}|$  featuring 1 in the  $i^{th}$  position if the segment  $LS$  has been annotated with the label  $l_i$ , 0 otherwise. Against this background, the concept of annotated segment is extended as follows:

**Definition 7** An **extended labeled segment**  $ELS$  is defined as a pair  $ELS = \langle S, \mathbf{v} \rangle$ , where  $S$  is a segment and  $\mathbf{v} \subseteq [0, 1]^{|\mathcal{L}|}$ .

Similarly, it is possible to extend the concept of labeled path into **Extended labeled path**.

The core of the approach proposed in this paper is a recommendation model capable of accommodating the user's preferences in order to suggest the most suitable paths. In order to do this, a compact and unambiguous representation of both user requirements and route characteristics is needed. Below is the definition of a vector representation of an extended labeled path. The definition proposes a normalized function that includes both the lengths of the individual segments and their specific properties. The resulting representation enables a fast comparison between different paths, using similarity metrics (such as the *cosine similarity*).

**Definition 8** Given an extended labeled path  $P$  consisting of  $n$  segments  $P = \langle (S_i, \mathbf{v}_i) | i = \{1, \dots, n\} \rangle$ , the **vector representation of path  $P$**  is formally defined as:

$$\mathbf{r}_P = \frac{1}{\text{len}(P)} \sum_{i=1}^n \text{len}(S_i) \cdot \mathbf{v}_i \quad (3)$$

where  $\text{len}(P)$  is the length of the path  $P$  and  $\text{len}(S_i)$  is the length of the  $i^{\text{th}}$  segment of  $P$ .

As for the user interests, they are also represented as a vector  $\mathbf{u}$  in a  $|\mathcal{L}|$ -dimensional space where each dimension  $i$  represents the level of interest shown by the user for the  $i^{\text{th}}$  feature of set  $\mathcal{L}$ . As a consequence, both the path representative vector and the user profile vector are in the same space. From a different perspective, the user’s profile vector can be seen as representative of the *ideal* path for that user (more details can be found in Section 4.2).

#### 4 A tale of two (labeled) cities

The ultimate goal of this section is to present the building blocks of our approach for the composition and recommendation of multi-feature paths, that is a comprehensive methodology that, we believe, covers all the essential aspects of this topic. Some of them have been addressed using techniques widely acknowledged in the literature, and others are original. Given the limited space available, the innovative aspects are primarily addressed, still briefly touching on the others.

Figure 1 exhibits the architecture and workflow of the prototype built to validate the approach. Two information flows are visible, along with the components that manage them: on the one hand, the prototype processes the routes labeled by the user by filtering and splitting them, finally generating a consolidated graph; on the other hand, the same information is used to update the user’s profile and recommend the paths that come closest to her needs. The following subsections discuss the two processes in depth.

##### 4.1 Path Processing

In the first phase of the path processing flow, the system acquires user labeled paths. Since trajectories recorded by GPS devices can be noisy and inaccurate (due to instruments precision), it is crucial to apply suitable filtering techniques: in our approach, the well-known SPT [18] algorithm is used.

The phase following the collection of labeled paths is their analysis and the generation of a *consolidated graph*. Such a graph (also referred to as *map*) is a unified view featuring all the possible paths uploaded in the system. A series of preprocessing operations are required to obtain the graph:



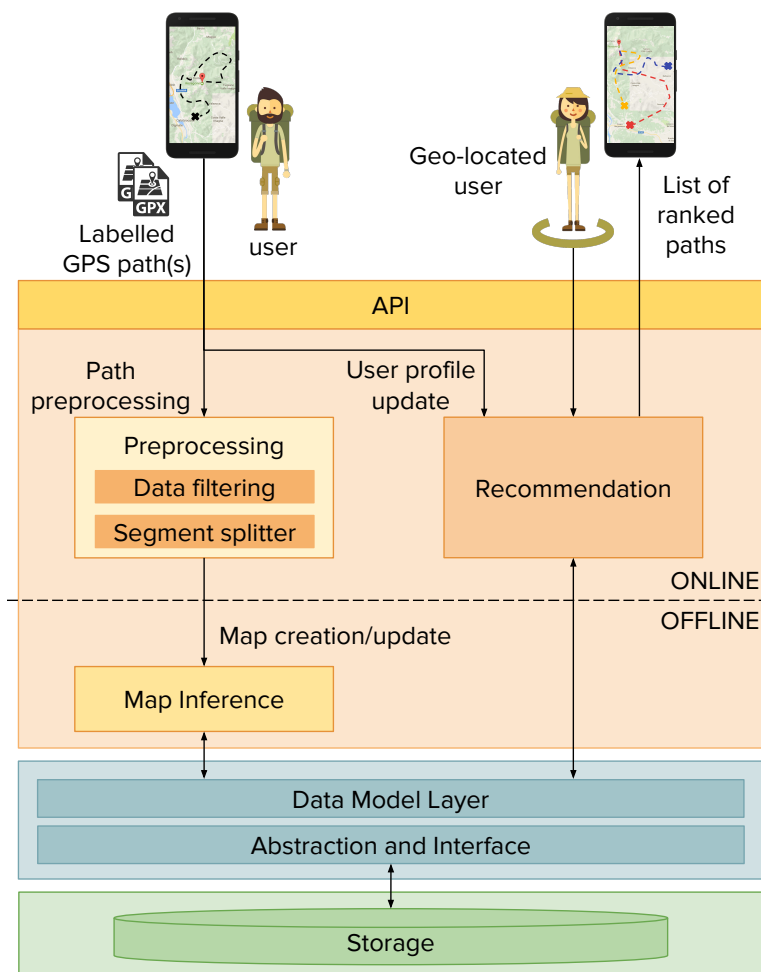


Fig. 1 General architecture of the prototype

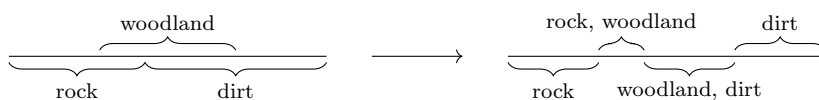
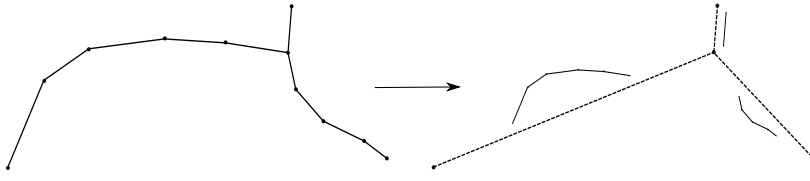


Fig. 2 Segment splitting and label regrouping based on user annotations

1. As user-loaded paths may contain partly overlapping labeled segments, it is necessary to split them into smaller but adjoining segments and regroup the labels (Figure 2 depicts this process).
2. An additional segmentation strategy (named *turning points*) is applied, as described in [27]; it generates a new segment whenever a turn angle (above a certain threshold) is detected.



**Fig. 3** Reduction of the consolidated graph

3. Finally, different paths can have shared sections, and these must be identified and merged. One of the main requirements of this approach is the ability to evolve, so it is of paramount importance that the consolidated graph is easy to update. For this reason, the solution proposed by [1] has been implemented. Indeed, being based on Fréchet's distance, and independent of the sampling frequency and the speed of movement of the objects, it allows to incrementally include new trajectories to the system, also recorded by different tracker devices.

The resulting consolidated graph contains a considerable number of edges and is likely to cause performance issues in the recommendation algorithm. For this reason, it undergoes a simplification process presented here for the first time that is performed in the form of a batch process (off-line). The outcome is a more compact graph, featuring only those nodes that in the consolidated graph are either junctions between different segments or source and destination points; moreover, if an edge between two nodes exists in the reduced graph, this implies the existence of a path between the same two points in the original graph that never forks.

The *simplification procedure* illustrated in Figure 3 is based on the identification of junctions, i.e., nodes with a degree higher than 2, and source and destination nodes (degree equal to 1). The graph is then simplified by identifying existing paths between pairs of nodes (either junction, source, and destination), passing through only degree-2 nodes, and replacing them with a single edge that connects the pair of nodes considered.

It is also essential to note that to provide the user with complete information, it is not possible to afford to lose such valuable information as the geometry and geolocation of the various segments. For this reason, the geometry of each original path is preserved, stored as a property of the new edge.

Similarly, the *consolidated graph assembling process* may result in a mismatch between the original trajectory segments, which are labeled, and the new segments appearing in the reduced graph (see Figure 4), yet this information content is essential for the recommendation and therefore it is essential to preserve it. Therefore, the following strategy capable of associating each label with a percentage of presence (*presence degree*) within a graph segment is proposed:

1. For each labeled segment, an area delimited by a symmetrical sifting of the segment upwards and downwards is identified (referred to as *buffer area*),

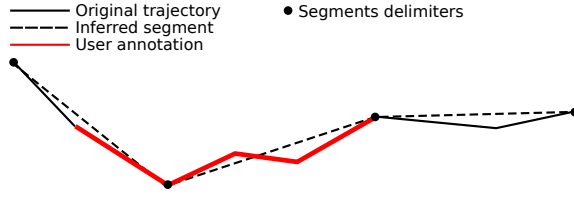


Fig. 4 Original path vs. inferred segments

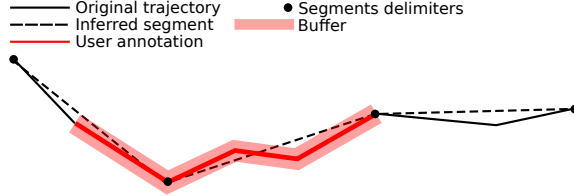


Fig. 5 Buffer of a labeled segment

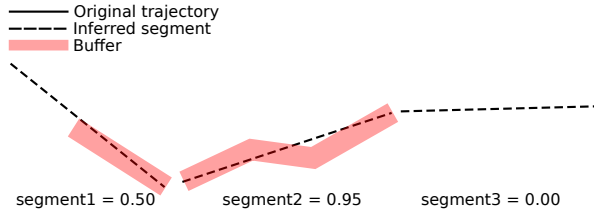


Fig. 6 Extended labeling process

see Figure 5. Note that the buffer width is a parameter that has been set based on the results of preliminary experiments;

- Given an edge  $a$  of the reduced graph featuring a labeled segment  $LS = \langle S, L \rangle$  (Definition 5), the *presence degree* of  $LS$  on  $a$  is calculated as the ratio between the length of the part of  $a$  contained in the buffer,  $a_{buf}$  and the length of  $S$ ,  $len(S)$ , that is  $d(a) = \frac{a_{buf}}{len(S)}$  (Figure 6).

This procedure enables to define for each edge of the reduced graph, an  $|\mathcal{L}|$ -dimensional vector  $\mathbf{v}$ , where the  $i^{th}$  dimension is the presence degree of the  $i^{th}$  label within the segment, which leads to having an extended labeled segment (Definition 7) for each edge. This information is used to compute a vector representation for each path in the consolidated graph (Definition 8).

The remainder of this section presents a short discussion on the complexity of the path processing to investigate the efficiency and the scalability of the approach on massive datasets. Notice that this phase has no strict time requirements and can be performed offline, as a batch process. Furthermore, the algorithm for the map inference implements an incremental track insertion procedure that, given a new trajectory, features a time complexity of  $O(e_u(\chi\psi)^2 \log(\chi + \psi))$ , where  $\chi$  and  $\psi$  are, respectively, the dimension of the graph before the update, and of the path, and  $e_u$  is the number of edges up-

dated [1]. Such complexity guarantees an acceptable level of scalability both in the case of creating the map from scratch<sup>4</sup> and in the case of updating an existing map. Lastly, the consolidated graph assembling process, instead, can be efficiently implemented by exploiting overlay functions available within the GIS system (state-of-the-art algorithms can compute overlays in  $O(w + y)$ -time, where  $w$  is the number of edges of the polygon and  $y$  the number of existing intersections [2]) whereas the simplification procedure is performed processing once all the nodes of the map,  $O(v)$ , where  $v$  is the number of nodes in the graph.

#### 4.2 Path Composition and Recommendation

The objective of a path recommender system is to return a list of the top-k interesting paths for a user. It is therefore essential to know the user’s preferences and keep them up to date. For this reason, upon subscribing, the system requires the user to answer a few questions to understand her preferences and initialize the profile vector. After that, the user profile vector is implicitly updated using the Rocchio’s Algorithm [22], which classifies interesting and non-interesting items for a user, and uses these sets for updating the user profile incrementally.

Furthermore, unlike other recommendation problems, when it comes to recommending routes, it is necessary to develop an algorithm that considers both the user’s profile and her specific and momentary needs as the starting point, the arrival point, and the length of the path. More specifically, given a user and a geographical position (*starting point*), the result of the recommendation phase must be a list of paths beginning in an area surrounding the starting point (*area<sub>start</sub>*). In addition, three possible usage scenarios are considered: in the first one the user indicates the desired length of the path, in the second one she indicates the point near which the path should end (*arrival point*), in the third, neither the length nor the arrival point is provided. In the latter scenario, a heuristic approach, based on recent activities of the target user and exploiting at the same time the recent activities of other users with similar profiles, is used to predict the suitable length for the paths to recommend.

More specifically, the heuristic first calculates the following two measures:

$$len_{SU} = \frac{\sum_{g=today}^{g_{max}} \gamma \cdot |SU_g| \cdot w\_median(SU_g)}{\sum_{g=today}^{g_{max}} \gamma \cdot |SU_g|} \quad (4)$$

$$len_U = \frac{\sum_{g=today}^{g_{max}} \gamma \cdot |U_g| \cdot median(U_g)}{\sum_{g=today}^{g_{max}} \gamma \cdot |U_g|} \quad (5)$$

where  $SU_g$  is a multi-set containing the length of all the paths recorded on day  $g$  by other users starting within the *area<sub>start</sub>*, whereas  $U_g$  contains

<sup>4</sup> The algorithm’s performance, in the original paper, was tested against a dataset containing 3237 vehicle trajectories, for a total of 57109 position samples, generating the underlying map within 11 minutes on a commodity machine.

the length of all the paths recorded on  $g$  by the target user. A *forgetting* mechanism is implemented by setting an oblivion point in the past, the day  $g_{max}$ . Moreover, while  $median(U_g)$  is the daily median of the lengths in  $U_g$ ,  $w\_median(SU_g)$  is the daily median of the lengths in  $SU_g$ , weighted on the similarity between the target user's vector profile and the one of the user  $u_i$  that uploaded the considered path.

Thus, Equations 4 and 5 compute the mean of the daily length medians, weighted by the following factors:

- The decay factor  $\gamma$ , which decreases the daily median when the considered date is less recent. This factor gives more importance to recent information;
- The number of paths  $|SU_g|$ , as the median is more reliable when computed on a larger number of paths.

Linearly combining Equations 4 and 5 leads to:

$$len_{ref} = (1 - \lambda) \cdot len_{SU} + \lambda \cdot len_U \quad (6)$$

Finally, the system considers only paths of length  $len(P) \in len_{bound}$ , where:

$$len_{bound} = [(1 - c) \cdot len_{ref}, len_{ref}], 0 < c < 1 \quad (7)$$

By means of the heuristics described above, the third scenario is reduced to the first one, whereas the second already represents a particular case of the latter. When an arrival point is given, the system will consider all the paths that end in an area surrounding the arrival point ( $area_{end}$ ).

The following is a description of the composition and recommendation algorithm for multi-feature paths, whose execution assumes as input a starting position and the length of the path (entered directly by the user or calculated by the heuristic). A thorough discussion of the first scenario is held, which also covers the third one, while details are provided only for those steps that require extensions in order to cover also the second scenario.

The algorithm is constituted by two separate phases: the path composition and the recommendation phases. In the former, given the reduced graph, the starting point and the length of the path, the algorithm identifies (if any) all the possible paths that verify the user's requests by following the graph and constructing the paths, progressively adding segments until the desired length is reached and avoiding generating cycles. When the arrival point is given instead of the length (second scenario), the system identifies all the possible paths between the starting and arrival points. In the latter, the representative vector  $\mathbf{r}_P$  is computed for each generated path and compared with the user's profile. The representation given in Equation 3 has been extended in order to investigate how the model is affected by the following aspects: (i) the similarity  $sim_l(\mathbf{v}, \mathbf{u})$  of each segment  $\mathbf{v}$  with respect to the user profile  $\mathbf{u}$  (named *local similarity*); (ii) the order of each segment in the path, assuming that it is preferable to have interesting features at the beginning of the path; (iii) the similarity  $sim_l(\mathbf{u}, \mathbf{r}_P)$  of the computed vector  $\mathbf{r}_P$  and the user profile  $\mathbf{u}$  (named *global similarity*); (iv) the length of the computed path, with respect to the

reference value  $len_{ref}$ ; in the second scenario, the distance between the arrival point selected by the user, and the arrival point of the path is considered.

In order to investigate aspects (i) and (ii) Equation 3 is extended as follows:

$$\mathbf{r}_P = \frac{1}{len(P)} \cdot \sum_{i=1}^n len(S_i) \cdot \mathbf{v}_i \cdot sim_l(\mathbf{u}, \mathbf{v}_i) \cdot \beta_i \quad (8)$$

where  $sim_l(\mathbf{u}, \mathbf{v})$  is a similarity measure,  $\mathbf{u}$  and  $\mathbf{v}$  are the user profile vector and the segment vector, respectively, and  $\beta_i$  is a decay factor. We set  $\beta_i = e^{-f(i)}$  as decay factor, where:

$$f(i) = \frac{1}{len(P)} \cdot \sum_{j=1}^{i-1} len(S_j)$$

so the farther the segment is from the origin, the lower turns out to be its weight. Thus, Equation 8 becomes:

$$\mathbf{r}_P = \frac{1}{len(P)} \cdot \sum_{i=1}^n len(S_i) \cdot \mathbf{v}_i \cdot sim_l(\mathbf{u}, \mathbf{v}_i) \cdot e^{-\frac{\sum_{j=1}^{i-1} len(S_j)}{len(P)}} \quad (9)$$

As a result, the algorithm can estimate the user score for a path by applying a global similarity measure, eventually weighted by considering aspect (iv):

$$score(\mathbf{u}, P) = \frac{len(P)}{len_{ref}} \cdot sim_g(\mathbf{u}, \mathbf{r}_P) \quad (10a)$$

Within the second scenario, Equation 10 is adjusted by considering the distance between the arrival point  $p_{end}$  selected by the user, and the arrival point  $P.p_n$  of the path, as follow:

$$score(\mathbf{u}, P) = \frac{sim_g(\mathbf{u}, \mathbf{r}_P)}{dist(p_{end}, P.p_n)} \quad (10b)$$

Since Equations 10a and 10b are interchangeable in the algorithm workflow, in the next we only refer to Equation 10.

The Algorithm 1 shows in the form of pseudo-code the implemented recommendation approach (leaving aside the creation of the consolidated graph) aiming at clarifying the role played by Equations 9 and 10.

Two distinct blocks are clearly recognizable in the code. In the first block (lines 4 to 8), all paths that meet the user's requirements (i.e., having the starting point in the surrounding area and a length value in the range set  $len_{bound}$ ) are identified and collected. The *findAllPathsFromSegment*( $G, s, lenRef$ ) function traverses the consolidate graph in order to combine all the possible paths that start from the selected segment  $s$  and which length is less than  $lenRef$ , avoiding cycles (i.e., each segment appears at most once in each path). The generated set is then filtered by removing paths that are shorter than the *minL* parameter.

**Algorithm 1:** Path recommendation

---

```

Data:
   $G$ : unified graph (map)
Input:
   $U$ : user vector profile
   $minL$ : path min length
   $lenRef$ : reference length
   $p$ : starting point
   $t$ : threshold
Output:
   $L$ : ordered list of path

  /* init variables */
1  $L \leftarrow []$ ;

  /* find all paths starting in the surrounding area */
2  $nearSegments \leftarrow getNearSegments(G, p, t)$ ;
3  $P \leftarrow \emptyset$ ;
4 foreach  $s$  in  $nearSegments$  do
5    $subPaths \leftarrow findAllPathsFromSegment(G, s, lenRef)$ ;
6   foreach  $p$  in  $subPaths$  do
7     if  $len(p) > minL$  then
8        $L \leftarrow L \cup \{p\}$ ;

  /* rank paths */
9 foreach  $p$  in  $L$  do
10    $V \leftarrow pathVector(p)$  // Equation 9;
11    $sim \leftarrow cosine(V, U)$ ;
12    $score \leftarrow \frac{sim \cdot length(p)}{lenRef}$  // Equation 10a;
13    $append(L, (p, score))$ ;
14 return  $sortedList(L)$ 

```

---

In the following block of code (lines 9 to 13), the representative vector for each path is computed alongside its overall score, based on the given user profile vector. Finally, the path list is sorted according to this score. Since Equations 9 and 10 are central to our approach, and it is worth noting that as  $sim_l$  and  $sim_g$  are generic similarity functions, the same two equations define a broad class of possible recommendation algorithms. The next section presents and assesses some of them.

To conclude, it is worth noticing that the path composition procedure mainly consists of a per-user graph traversal on the consolidated graph, which can be implemented efficiently (time complexity of  $O(v + e)$ , where  $v$  and  $e$  are the number of nodes and the edges of the graph, respectively) and even parallelized to be executed on demand [19]. The recommendation algorithm, on the other hand, computes and sorts representative vectors of the paths, which can be performed in  $O(n_p |\mathcal{L}| n_{max})$ , where  $n_p$  is the number of paths,  $|\mathcal{L}|$  the number of features, and  $n_{max}$  the maximum number of segments in the paths, and  $O(n_p \log n_p)$ , respectively.

## 5 Experimental Validation

Some initial assessments of the quality of the presented approach and the validity of the underpinning hypotheses are presented in this Section. Experimenting with path-based recommender systems can be a cumbersome task as it often requires tracking real users activity and this is particularly problematic unless you can provide a system with a suitable Technology Readiness Level. Since this is still a fresh field in literature and no gold standards are available to the best of our knowledge, different researcher groups resorted to prove the soundness of their models involving users via surveys [13]. This paper follows this *modus operandi* creating a control group and requiring its components to express their preference for different multi-feature paths.

The rest of this section is structured as follows: in subsection 5.1 the experimental setup is described in detail, the models compared are described in subsection 5.3, and the results are presented and discussed in subsection 5.4.

### 5.1 Design of the experiment

This section outlines the most significant details of the experiment carried out. Firstly, the prototype system has been fed with 16 real hiking trajectories (retrieved from *Wikiloc*) recorded in a small area in the north of Italy, featuring lengths spread from a minimum of 3.83km to a maximum of 18.59km (the average length is 9.25km). The outcome of this process is the consolidated graph underlying the trajectories, which contains 124 unique segments. Four possible annotations have been then considered (namely, rock, dirt, woodland, asphalt, plus an extra annotation that represents the void annotation) and assigned to the different segments of the graph using a stochastic procedure. As a result, 11% of segments are annotated as rock, 30% as dirt, 26% as woodland, 3% as asphalt and 45% as none (void annotation). 16% of segments are annotated with two labels.

Then the consolidated graph has been explored to generate a list of paths via composition. Such a list contains all the routes that start in the area of  $7850m^2$  surrounding an arbitrary point, and with a length of between 3500 and 4000 meters; the full list contains 26 real paths that combine a total of 36 segments, the average number of segments per path is 8.65, while the minimum and maximum numbers of segments per path are 5 and 12, respectively. Notice that the selected paths share some segments with an average reuse ratio of 0.33 (this value has been achieved by omitting the first segment, which is shared among every path). We can confidently state that the selected paths represent a good sample since the annotations of their segments have a distribution resembling that of the full dataset (17% rock, 30% dirt, 25% woodland, 2% asphalt, 47% none, 19% with two labels).

Subsequently, a set of users active in online social network communities have been engaged in taking part in a survey. At the end of the data gathering phase, 38 questionnaires have been collected. The users have been asked for



**Table 1** Experimental setup

$area_{start}$	7850m <sup>2</sup>	Users	38
$len_{bound}$	[3500m, 4000m]	Paths	26
$sim_l, sim_g$	cosine similarity	$ \mathcal{L} $	4

their age, occupation, interests for outdoor activities, and frequency of those activities in the previous month. Their preferences with respect to the path characteristics have been also recorded to allow the system to build a profile vector for each user. Such a vector representation features 4 components, whose values are normalized so that the sum is equal to 1.

Additionally, users have been asked to express a preference score (from 0 to 5) to each path of the list. It may seem that the number of considered paths, i.e., 26, is somewhat limited, yet it is worth considering that the evaluation process requires substantial effort, and long evaluation tasks can introduce fatigue biases in the experiments [8]. At the end of the process, 38 preference rankings have been collected, which represent our ground truth. Finally, notice that the inter-rater agreement among all users has been quantified using the Kendall’s W, achieving a value of 0.112, which is evidence of a low degree of agreement and justifies the need for a personalized per-user list of paths. Table 1 shows a summary of the experimental settings. The source code, models and datasets used for the experiments are available on GitHub<sup>5</sup>.

## 5.2 Evaluation Metric

To assess the quality of the recommendations spawn with respect to the user’s preferences, the nDCG [12] metric has been employed. This measure is useful to evaluate rankings: given a ranked list and a reference list, the nDCG estimates its quality by giving more importance to items in a higher position of the ranked list with respect to their rankings in the reference list. The nDCG can be computed by considering lists of different length: in the following nDCG@k denotes the nDCG computed on the top-k elements, which is useful to compare algorithms in different scenarios by changing the value of k. It also has a realistic interpretation: a user generally does not want to explore hundreds of item, but she would like to find the paths in the top returned items (generally 5 or 10). For the sake of completeness, the procedure to compute the nDCG is reported below. The DCG must be computed first:

$$DCG_k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (11)$$

where  $rel_i$  is the score given by a user with respect to a given path. The nDCG is thus computed as:

$$NDCG_k = \frac{DCG_k}{IDCG_k} \quad (12)$$

<sup>5</sup> <https://github.com/vcutrona/paths-rs>

where the  $IDCG_k$  is the ideal DCG. By considering all the scores given by the user to the paths, the IDCG is achieved by computing the DCG on the list of paths ordered in decreasing order with respect to the score. Clearly, this results in the best possible ranking to the user and in that case the nDCG would be equal to 1.

### 5.3 Tested Recommendation Models

Different variations of the core model have been implemented to assess the soundness of the proposed approach and validate or reject the underpinning assumptions. Firstly, three main algorithms have been implemented, which combine global and local similarities in different ways:

1. The first approach (named **Local**) relies only on the local similarity ( $sim_l$ ) of each segment with respect to the user's profile, without computing the global similarity. In this way, it can be observed whether the average user favors the assessment of individual (*local view*) segments over considering the entire route (*global view*). This algorithm has been obtained by modifying Equation 9 as follows:

$$r_P = \frac{1}{n \cdot len(P)} \cdot \sum_{i=1}^n len(S_i) \cdot sim_l(\mathbf{u}, \mathbf{v}_i) \cdot e^{-\frac{\sum_{j=1}^{i-1} len(S_j)}{len(P)}}$$

and Equation 10 as:

$$score(\mathbf{u}, P) = \frac{len(P)}{len_{ref}} \cdot r_P$$

2. The second algorithm has been created with the intention of validating the alternative assumption, that is to test if users are more focused on the overall path, ignoring the relative importance of each segment. Indeed, this variation (named **Global**) considers only the global similarity ( $sim_g$ ) of the path representative vector with respect to the user's profile, neglecting the local similarity. This model alters Equation 9 as follows:

$$\mathbf{r}_P = \frac{1}{len(P)} \cdot \sum_{i=1}^n \mathbf{v}_i \cdot len(S_i) \cdot e^{-\frac{\sum_{j=1}^{i-1} len(S_j)}{len(P)}}$$

while the path score is still computed via Equation 10.

3. As the third variation, the aforementioned assumptions have been combined into a new model (named **G-Local**) that considers both the local and the global similarity measures. The hypothesis underlying this model is that users appreciate the global result of the compositional task, but are also interested in the relative importance of each segment.  $\mathbf{r}_P$  and  $score(\mathbf{u}, P)$  can be computed as described in Equations 9 and 10, respectively.

Secondly, in order to validate the study hypotheses that (i) it is meaningful to consider the relative length of each segment with respect to the total path length, and (ii) the user’s satisfaction also hinges on the order in which segments appear along the path, three relaxed models have been conceived that do not implement those assumptions (named **Local Relaxed**, **Global Relaxed**, **G-Local Relaxed**), which are derived directly from the previous three models by imposing  $len(P) = 1$  and  $len(S_i) = \beta_i = 1 \forall i \in \{1, \dots, n\}$ . Lastly, a random algorithm (named **Random**) that does not exploit path semantics has been established as a baseline for the experiment.

#### 5.4 Results analysis

Each recommendation model has been applied to the profiles in the control group thus generating 38 separate rankings for the 26 paths considered in the experiment. The resulting rankings have then been compared against the related user’s ideal path list, and the nDCG indicator has been computed. The nDCG value has been then averaged over the components of the group. As for the Random baseline, however, it was necessary to proceed differently, as its outcome is non-deterministic, it was necessary to repeat the generation of the list of recommendations a sufficient number of times to stabilize the value of the nDCG indicator average.

In order to gain the most in-depth insight out of the experiment, both the nDCG@5 and the nDCG@10 have been computed, which consider the top-5 and top-10 path recommendations, respectively. Behind this choice is the desire to ascertain whether the effectiveness of the various models varies as the number of routes to be recommended increases. Furthermore, in addition to the control group in its entirety, the users have been further categorized based on the interests expressed in the survey; two additional categories have been identified: (i) the *interested users*, people interested in outdoor activities, and (ii) the *expert users*, interested users that also practice outdoor activities frequently ( $\geq 4$  times in a month). The interested users are 22, while the experts are 7.

Table 2 shows the results of our recommendation models for 7 algorithms, 2 indicators, and 3 categories of users. It is apparent that all algorithms based on the presented approach outperform the baseline, and this seems to confirm that users need a customized path recommendation, tailored to their interests. Moreover, even if the number of considered users decreases, one can observe that in general, the models achieve better results for interested and expert users categories, when compared with the all users category. Finally, it is interesting that the algorithm based on the Global model outperforms the others, leading us to suppose that users tend to evaluate a path in its entirety rather than the individual segments that make it up.

**Table 2** nDCG for the different approaches

	Interested Users		Expert Users		All Users	
	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10
G-Local Relaxed	0.7817	0.7965	<b>0.7992</b>	0.7958	0.7715	0.7964
Local Relaxed	0.8068	0.8083	0.7887	0.7818	<b>0.8036</b>	0.8066
Global Relaxed	0.8004	0.8124	0.7918	0.7919	0.7762	0.7982
Random	0.7189	0.7454	0.6939	0.7195	0.7421	0.7682
G-Local	0.7547	0.7706	0.7723	0.7583	0.7473	0.7732
Local	0.7981	0.8190	0.7909	0.8009	0.7786	0.8069
Global	<b>0.8167</b>	<b>0.8298</b>	<b>0.7992</b>	<b>0.8103</b>	0.7979	<b>0.8194</b>

## 6 Conclusions and Future Work

In this work, an original approach for path composition and recommendation has been presented and discussed. It exploits both the compositional nature of trajectories and semantic annotations to generate a path consolidated graph, which allows the system to infer new paths and semantic-based recommendations. A prototype implementation of the proposed approach has been presented and the workflow discussed in details. In particular, the defined approach is able to process raw trajectories to create a compact, unified representation that, in turn, is exploited to scout and compose new routes to be recommended to the final users. The underlying recommendation models have also been presented, discussed and evaluated by means of an experiment involving real users.

Future work includes the extension of the presented approach in several directions, among which the extension of the core recommendation model with new variables such as the altitude with the aim of providing a more accurate recommendation. Another interesting direction of study concerns the definition of models and methodologies to provide adaptive recommendations suggesting path variations in real time as new information becomes available (e.g., the weather changes while a user is skiing).

## References

1. Ahmed, M., Wenk, C.: Constructing street networks from gps trajectories. In: Algorithms – ESA 2012: 20th Annual European Symposium, 2012. Proc., pp. 60–71. Springer Berlin Heidelberg (2012)
2. Audet, S., Albertsson, C., Murase, M., Asahara, A.: Robust and efficient polygon overlay on parallel stream processors. In: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, pp. 304–313 (2013)
3. Aung, H.H., Guo, L., Tan, K.L.: Mining sub-trajectory cliques to find frequent routes. In: Advances in Spatial and Temporal Databases: 13th International Symposium, SSTD 2013. Proc., pp. 92–109. Springer Berlin Heidelberg (2013)
4. Baraglia, R., Frattari, C., Muntean, C.I., Nardini, F.M., Silvestri, F.: A trajectory-based recommender system for tourism. In: Active Media Technology: 8th International Conference, AMT 2012. Proc., pp. 196–205. Springer Berlin Heidelberg (2012)
5. Cai, G., Lee, K., Lee, I.: Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. *Expert Systems with Applications* **94**, 32 – 40 (2018)

6. Cao, L., Krumm, J.: From gps traces to a routable road map. In: Proc. of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, pp. 3–12 (2009)
7. Chen, C., Lu, C., Huang, Q., Yang, Q., Gunopulos, D., Guibas, L.: City-scale map creation and updating using gps collections. In: Proc. of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pp. 1465–1474 (2016)
8. Choi, B.C.K., Pak, A.W.P.: Bias, overview. *Encyclopedia of Biostatistics* **1**, 416–423 (2005)
9. Cui, G., Luo, J., Wang, X.: Personalized travel route recommendation using collaborative filtering based on gps trajectories. *International Journal of Digital Earth* **11**(3), 284–307 (2018)
10. Davies, J.J., Beresford, A.R., Hopper, A.: Scalable, distributed, real-time map generation. *IEEE Pervasive Computing* **5**(4), 47–54 (2006)
11. Edelkamp, S., Schrödl, S.: Route planning and map inference with global positioning traces. In: *Computer Science in Perspective: Essays Dedicated to Thomas Ottmann*, pp. 128–151. Springer Berlin Heidelberg (2003)
12. Gunawardana, A., Shani, G.: Evaluating recommender systems. In: *Recommender Systems Handbook*, pp. 265–308. Springer US (2015)
13. Herzog, D., Massoud, H., Wörndl, W.: Routeme: A mobile recommender system for personalized, multi-modal route planning. In: Proc. of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP '17, pp. 67–75 (2017)
14. Karagiorgou, S., Pfoser, D.: On vehicle tracking data-based road network generation. In: Proc. of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12, pp. 89–98 (2012)
15. Kitayama, D., Ozu, K., Nakajima, S., Sumiya, K.: A route recommender system based on the user's visit duration at sightseeing locations. In: *Software Engineering Research, Management and Applications*, pp. 177–190. Springer International Publishing (2015)
16. Lee, J.G., Han, J., Whang, K.Y.: Trajectory clustering: A partition-and-group framework. In: Proc. of SIGMOD International Conference on Management of Data, SIGMOD'07, pp. 593–604 (2007)
17. Liu, H., Schneider, M.: Similarity measurement of moving object trajectories. In: Proc. of the Third ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '12, pp. 19–22 (2012)
18. Meratnia, N., de By, R.A.: Spatiotemporal compression techniques for moving point objects. In: *Advances in Database Technology - EDBT 2004: 9th International Conference on Extending Database Technology*, pp. 765–782. Springer Berlin Heidelberg (2004)
19. Merrill, D., Garland, M., Grimshaw, A.: Scalable gpu graph traversal. *SIGPLAN Not.* **47**(8), 117–128 (2012)
20. Nardini, F.M., Orlando, S., Perego, R., Raffaetà, A., Renso, C., Silvestri, C.: Analysing Trajectories of Mobile Users: From Data Warehouses to Recommender Systems, pp. 407–421. Springer International Publishing (2018)
21. Socharoentum, M., Karimi, H.A.: Multi-modal transportation with multi-criteria walking (mmt-mcw): Personalized route recommender. *Computers, Environment and Urban Systems* **55**, 44 – 54 (2016)
22. Wang, C., Shen, Y., Yang, H., Guo, M.: Improving rocchio algorithm for updating user profile in recommender systems. In: *Web Information Systems Engineering – WISE 2013: 14th International Conference, Proc., Part I*, pp. 162–174. Springer Berlin Heidelberg (2013)
23. Yin, P., Ye, M., Lee, W.C., Li, Z.: Mining gps data for trajectory recommendation. In: *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014. Proc., Part II*, pp. 50–61. Springer International Publishing (2014)
24. Ying, J.C., Chen, H.S., Lin, K.W., Lu, E.H.C., Tseng, V.S., Tsai, H.W., Cheng, K.H., Lin, S.C.: Semantic trajectory-based high utility item recommendation system. *Expert Systems with Applications* **41**(10), 4762 – 4776 (2014)
25. Yoon, H., Zheng, Y., Xie, X., Woo, W.: Smart itinerary recommendation based on user-generated gps trajectories. In: *Ubiquitous Intelligence and Computing: 7th International Conference, UIC 2010. Proc.*, pp. 19–34. Springer Berlin Heidelberg (2010)

- 
26. Zhang, Y., Siriaraya, P., Wang, Y., Wakamiya, S., Kawai, Y., Jatowt, A.: Walking down a different path: Route recommendation based on visual and facility based diversity. In: Companion Proc. of the The Web Conference, WWW '18, pp. 171–174 (2018)
  27. Zheng, Y.: Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.* **6**(3), 29:1–29:41 (2015)
  28. Zheng, Y., Xie, X.: Learning travel recommendations from user-generated gps traces. *ACM Trans. Intell. Syst. Technol.* **2**(1), 2:1–2:29 (2011)