Personalised Recommendations for Modes of Transport: A Sequence-based Approach

Gunjan Kumar, Houssem Jerbi, and Michael P. O'Mahony
Insight Centre for Data Analytics
School of Computer Science, University College Dublin, Ireland
{firstname.lastname}@insight-centre.org

ABSTRACT

In this paper we consider the problem of recommending modes of transport to users in an urban setting. In particular, we build on our past work in which a general framework for activity recommendation is proposed. To model the personal preferences and habits of users, the framework uses a sequence-based approach to capture the order as well as the context associated with user activity patterns. Here, we extend this work by introducing a machine learning approach to learn and take into account the natural variations in the regularity and repetition of individual user behaviour that occur. We demonstrate the versatility of our recommendation framework by applying it to the transport domain, and an evaluation using a real-world (mode of transport) dataset demonstrates the efficacy of the approach.

1. INTRODUCTION

As digital technologies become ubiquitous in society, many aspects of our lives can now be passively recorded in digital formats. For example, locations visited, media consumed, physical activities performed, and modes of transport taken by users can be recorded using mobile devices. Such recordings can be used to generate detailed traces (or *lifelogs*) of an individual's life and can help in the design of bottom-up solutions to improve the quality of life for individuals, the urban environment and the efficiency of cities' operations systems [47]. Our work belongs to an emerging category of real-time recommender systems facilitated by such data, which are capable of generating recommendations at the right time and in the right way for a given user and context.

In previous work, we proposed a general framework for sequence- and context-based *activity* recommendation [23], where activities can take the form of daily tasks to complete, for example, or which music or web sites users should listen to or visit. In this paper, we apply our approach to recommend the next *mode of transport* that users should take in their journeys. This can help users to better plan their days, facilitate travel, and help service providers to better cater to the needs of the community.

A related application domain is mobile personal assistants, such as Google Now and Microsoft Cortana, which seek to show users the right information at the right time, without any query input [1, 19]. For example, these applications proactively show the estimated time of travel to home at the end of a working day; however, when using these applications, users need to preselect a particular mode of transport in advance. We believe that the utility of such applications would be significantly enhanced if the most suitable modes of transport to take were also recommended to users, and the provision of such a recommender system provides a practical rationale for this work

An inherent characteristic of the kinds of data captured by digital devices is their oftentimes inherently sequential nature, i.e. activities are performed in a particular order, and each activity in turn may influence the subsequent activities to be performed. Moreover, such activities are typically associated with multiple features or contextual data, such as location, time, weather, etc. It seems clear that the order encoded in such sequence data, along with the rich contextual data available, capture important information when it comes to modeling the preferences and personal habits of users. Indeed, it is known that in many instances users perform activities in patterns, sometimes consciously while often without conscious thought.

Traditional recommender systems, however, typically do not consider the order in which users perform activities and there is little work in urban computing which consider sequences and contextual data simultaneously. In this paper, we leverage such sequences and contextual data and show how they can be effectively used for recommending modes of transport to users. The main contributions of this work can be summarised as follows:

- A novel content-based approach for recommending the next activity (i.e. mode of transport) to users.
- Extending the recommendation framework in [23] by proposing new approaches to extract and match subsequences drawn from the past activity patterns (timelines) of users.
- A machine learning approach to learn the optimal subsequence length to be used when matching current and past subsequences of user activity patterns. In particular, given the variations in the regularity and repetition of user behaviour, our approach learns a personalised subsequence length for each user.
- The application of our recommendation framework to the transport domain. Experiments using a mode of transport dataset demonstrate that good quality recommendations are made for users, irrespective of their transport usage patterns, and likewise for modes of transport which are more or less common in the data.

The paper is organised as follows. Related work is discussed in Section 2. Sections 3 and 4 present our recommen-

UrbComp'16, August 14, 2016, San Francisco, USA

dation framework and classification approach, respectively. A comprehensive evaluation of our approaches is performed in Section 5, and conclusions are drawn in Section 6.

2. RELATED WORK

This work presents a recommendation approach which models both sequence and contextual information in order to generate activity recommendations for users. While related work in the urban computing [4, 47, 53] and recommender systems [3, 5, 31] domains exist which consider sequence and context individually, there appears to be little work which considers both together. In what follows, a review of some of the relevant related work in these areas is presented.

In urban computing research, one approach to capture sequence and geographical hierarchies in location trajectories is presented in [25], in which a hierarchical-graph-based model is described. This is further enhanced in [52] by modeling location popularity and user experiences to mine popular travel sequences across users in a non-personalised manner. Similarly, graph-based models have also been used to capture travel sequences for collaborative itinerary recommendation [43], i.e., recommending trips for a given start time, destination and duration of trip. However, these approaches do not leverage the context information associated with user's visits to locations for modeling purposes.

A popular approach for modeling sequence data in general has been Markov-based models. However, as the Markov assumption does not hold in many cases (e.g. web navigation), all-k^{th}-order Markov models have been used [29, 14] to capture higher order information. Inline with this observation, our recommendation approach does not rely on the Markov assumption. Furthermore, popular k^{th}-order Markov models, such as those used in [35, 6], consider elements of the sequence as atomic entities and are not suitable for sequences of activities with multiple features or context. For other approaches related to sequence-based web page and music recommendation, see for example [7, 9, 20].

The important role of context has frequently been noted in the literature. For example, Zheng et al. [44, 45, 46] proposed algorithms based on collaborative filtering which use location as context for recommending suitable activities. In this regard, a user-location-activity ratings tensor is constructed from information extracted from users' GPS trajectories, associated comments and additional information such as POIs, correlation between activities, etc. A collective tensor and matrix factorization model is then used to predict missing ratings or formulate the pairwise preferences of users for location-activity pairs in order to generate ranked lists of recommendations. An extension of this work using Higher Order Singular Value Decomposition is presented in [38]. However, such tensor-based models and other approaches to context-aware recommender systems [2, 40] do not capture sequence information. Moreover, context-aware recommender systems are typically reactive in the sense that recommendations are made for the current context; e.g., activities are recommended to users given their current location. In contrast, our approach is proactive as it recommends the *next* activity given the current context, i.e., the recommendation and the current context are temporally apart.

While capturing sequence information along with the context in user models has been suggested [3, 46] to improve recommendation, there are very few works which capture sequence and context simultaneously. In [23], we presented a content-based framework for recommending the next activity to the user based on the past sequences of activities performed and their associated features. We introduced a two-level distance measure to assess the similarity between sequences. In this paper, we extend our framework by introducing the concept of *timeline matching* and propose a classification approach to personalise matching for each user. This year, another interesting approach to capture both sequence and contextual information is presented [37]. This is achieved by modeling contextual information as stochastic processes and representing user data as time-series. The authors propose to capture temporal structure and co-movements of contextual information as latent factors. Interestingly, this approach is also proactive as it predicts the future intentions of users.

In order to characterise sequences, it is essential the capture the regularity in sequence patterns. Most recommender systems using sequences make use of n-grams to characterise sequences; however, n-gram based metrics are better at capturing repetition rather than regularity. In this work, we use *sample entropy*, proposed by Richman and Moorman [32], as a statistic to quantify the amount of regularity in data. Sample entropy is a modification of approximate entropy [28] and is closely related to Kolomogorov-Sinai (K-S) entropy [16, 18]. This statistic relates to quantifying the rate of information generation in the time-series. Sample entropy has been previously used to quantify regularity in physiological and biological time-series such as heart rate from ECG and brain activity from fMRI [11, 13, 27, 36]. However, to the best of our knowledge, it has not been used for the purpose of recommendation. Here, we use sample entropy based attributes for numerical as well as for categorical sequences, in order to build a learning model to personalise sequence matching and making recommendations for individual users.

Finally, we note that much of the research relating to modes of transport is on the classification of the mode of transport being used based on GPS, accelerometer, GIS and other assisted data [17, 34, 48, 49, 50]. Moreover, while activity recommendation is an emerging area of research, especially in location-based social networks [4], the focus of this work (recommending modes of transport for users to take in their journeys) remains a largely unexplored task.

3. RECOMMENDATION APPROACH

In this section, we formulate the problem of activity recommendation. Here, the activities under consideration are modes of transport and the objective is to recommend to the user the next mode of transport to take in their journey. We present our content-based recommendation algorithm along with the concept of matching unit to better capture the similarities between users' current and past activity patterns.

3.1 Problem Formulation

Our work is motivated by the assumption that people tend to repeat similar patterns of activities under similar circumstances. For example, a given user might have a habit of travelling by bus to a movie theatre on Saturday evenings, followed by dinner and commute by taxi to home. In such patterns of activities, the *order* is critical to the meaning of the sequence; for example, commuting to home followed by dinner is semantically different from dinner followed by commute to home. It is then important to detect similar patterns of activities in the user's past sequences for effec-

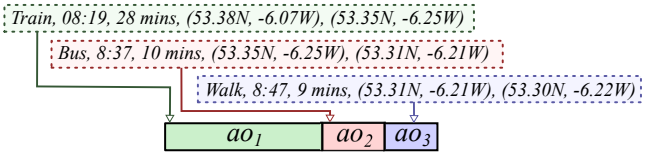

Figure 1: An example timeline consisting of three activity objects showing a commute from Howth to UCD in Dublin.

tively inferring the next activity that the user is likely to perform. Moreover, a critical determinant for inferring the next activity to perform is the surrounding context. For example, features such as the time of day, location and weather can determine if the user takes a bus or bike as the mode of transport. The circumstances impacting the activities can be captured through a set of *features*, f_1, ... , f_m, associated with each occurrence of an activity. A key feature of our work is that the sequence as well as the features associated with previous activity occurrences are both taken into consideration to recommend the next activity to the user.

3.2 Activity Timeline

We introduced the concept of an *activity object* and an *activity timeline* in [23]. An activity object refers to a single occurrence of an activity and consists of a set of features describing the activity or the context surrounding that particular occurrence of the activity. In this work, an activity refers to the act of taking a mode of transport.

When the set of activity objects that are performed within a time interval are arranged in a chronological order, the log of user activities represents a user *activity timeline* (or *timeline* for short). Formally, a user's activity timeline $\mathcal{T}$ is a chronological sequence of n activity objects that are performed during a time interval δ:

$$\mathcal{T} =< ao_1, ao_2, ..., ao_n > , \quad (1)$$

where activity object ao_i represents the i^{th} activity object performed by the user during δ (the duration between the starting time of ao_1 and the end time of ao_n). Further, each activity object is characterised by a set of m feature values:

$$ao_i = \left\{v_i^1, v_i^2, ..., v_i^m\right\} , \quad (2)$$

where v_i^j is the value of the feature f_j in activity object ao_i. In particular, the first feature denotes the *name* of the activity object; examples of such names include *bus*, *taxi*, *train*, etc. in the transportation domain and *working*, *commuting*, *socialising*, etc. in the domain of daily activities performed by users. The remaining features are domain dependent.

For example, Figure 1 shows an example timeline in the transportation domain which is composed of three activity objects. Each object is characterised by four features: *name (mode of transport, e.g. train, bus, walk etc.), start time, duration, start geolocation* and *end geolocation*[1].

3.3 Recommendation Algorithm

A key step of the activity recommendation process is to select past timelines that have similar patterns to the user's most recently performed activities.

For each user, the recommendation algorithm proceeds as follows. The task is to recommend the next activity to perform at a given point in time. Let ao_c denote the *current activity object*, i.e. the most recent activity performed by the user. A subsequence of activity objects, ending with ao_c, is extracted from the user's timeline; this subsequence is referred to as the *current timeline*, $\mathcal{T}_c$. The number of activity objects in $\mathcal{T}_c$ is given by the *matching unit* (mu).

For each previous occurrence in the user's timeline of an activity with the same name as ao_c (e.g. *train*, *bus*, *walk* etc.), a *candidate timeline* ($\mathcal{T}_j$) of length mu is extracted (see Figure 2). The subsequent activity object (denoted as ao_{rec}^j), which occurs immediately after $\mathcal{T}_j$, is then recommended and scored as follows. First, the distance $\left(d(\mathcal{T}_j, \mathcal{T}_c)\right)$ between the candidate timeline and the current timeline is computed using a two-level edit distance metric (see Section 3.3.1). A score is then computed as per Equation 3:

$$Score(ao_{rec}^j) = 1 - \frac{d(\mathcal{T}_j, \mathcal{T}_c) - \min\limits_{\mathcal{T}_p \in \mathscr{T}} d(\mathcal{T}_p, \mathcal{T}_c)}{\max\limits_{\mathcal{T}_p \in \mathscr{T}} d(\mathcal{T}_p, \mathcal{T}_c) - \min\limits_{\mathcal{T}_p \in \mathscr{T}} d(\mathcal{T}_p, \mathcal{T}_c)} , \quad (3)$$

where $\mathscr{T}$ is the set of candidate timelines.

Given a set of recommended activity objects (one from each candidate timeline), a list of activity names, ranked in descending order by the sum of the scores of the recommended activity objects in which they occur, is returned.

3.3.1 Distance between Timelines

For the purpose of determining the similarity between two timelines $\mathcal{T}_1$ and $\mathcal{T}_2$, the two-level similarity algorithm proposed in our earlier work [23] is used. This algorithm first rearranges the activities to achieve the same activity sequence and then aligns the values of the features of the corresponding activity objects. In the first step, the edit distance between the two timelines $\mathcal{T}_1$ and $\mathcal{T}_2$ is computed as the minimum cost of edit operations needed to transform the sequence of activities of $\mathcal{T}_1$ into the sequence of activities of $\mathcal{T}_2$. The second step involves the alignment of the feature values (i.e. *start time, duration, start geolocation* and *end geolocation*) of the corresponding activity objects in the two timelines. See [23] for further details on this approach.

3.3.2 Matching Unit

The matching unit has a critical role to play when finding similar patterns of activities in timelines, since it determines the length of the subsequences to be considered when calculating the distances between them. As such, the optimal matching unit for each user will differ, depending on the degree of repetition and regularity of activities performed.

In our previous work [23], current and candidate timelines were extracted on a daywise basis, i.e. timelines consisted of all activity objects starting from the beginning of a day and ending at the object with the same name as the current activity object, ao_c. We refer to this approach of matching timelines as *daywise* matching. Here, we introduce two other matching approaches, *N-count* and *N-hours* matching.

In the N-count matching approach, the N activity objects in the timeline preceding the current activity object form the current timeline (and likewise for candidate timelines). In the N-hours matching approach, all activity objects which occur during a specific timeframe before the current activity object are included in the current timeline (and similarly for candidate timelines). In this paper, due to limitations of space, we focus on N-count matching since experiments have shown that it outperforms the N-hours matching approach.

[1]Start and end geolocations are represented by latitude and longitude coordinates.

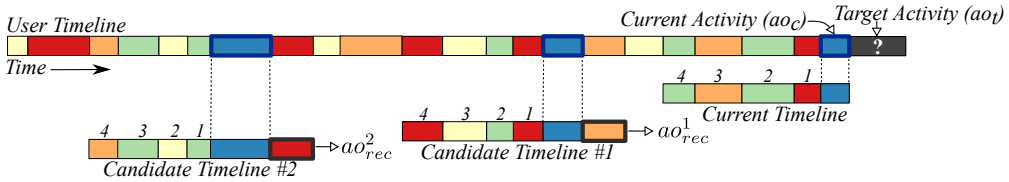

Figure 2: Overview of the recommendation approach (using N-count matching with $N = 4$). Different modes of transport (activities) in the user timeline are represented by coloured boxes.

4. LEARNING PERSONALISED OPTIMAL MATCHING UNITS

One of the essential steps in our approach is timeline matching, which requires the selection of a matching unit. In the context of N-count matching, this translates into selecting a value for N, i.e., the number of activities to be included in the extracted current and candidate timelines.

While the optimal value of N can be determined empirically for each user, such an approach is unlikely to be feasible in practice. Thus, we present a supervised classification approach to learn the optimal (from a recommendation accuracy perspective) value of N, i.e. N', for each user. However, given the natural variation in the activity patterns of users, learning an *exact* value for N' for each user may result in overfitting. Consequently, our proposed approach is to learn a range of values $\mathcal{N}'$ within which N' is likely to lie for each user. In this work, we consider three such ranges, and the classification task is to learn the optimal range for each user. Given a predicted range, N' is then set to a fixed value from within this range.

4.1 Attribute Extraction

Following the classical learning paradigm, each user is represented by an attribute vector, which is used to train a model to learn the optimal matching range. Attributes are extracted from the timeline of each user, and are selected to model the characteristics of each user's activity patterns.

Timeline Decomposition. As described in Section 3.2, a timeline $\mathcal{T}$ consists of a sequence of n activity objects, where each activity object is represented by a set of m features, $\mathcal{F}$. $\mathcal{F}$ is the set of features that describe the context associated with timeline activities. In order to extract attributes from timelines, it is useful to also consider the timeline from the perspective of each individual feature separately.

Thus, a timeline $\mathcal{T}$ can be decomposed into a set of $m \in \mathcal{F}$ different *feature-sequences*, $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_m$, one for each of the m features, with each new feature-sequence consisting of n elements. That is, given $\mathcal{T} =< ao_1, ao_2, ..., ao_n >$ and $ao_i = \{v_i^1, v_i^2, ..., v_i^m\}$, then $\mathcal{T} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_m\}$, where $\mathcal{S}_z = < v_1^z, v_2^z, ..., v_n^z >$ is the feature-sequence for the z^{th} feature of the activity object.

Linear Mapping of Feature Sequences. For some attributes, each element v_i^z of a feature sequence $\mathcal{S}_z$ is given by a single (1-dimensional) value (e.g. the feature activity *duration*). However, in the case of features *start geolocation* and *end geolocation*, elements are represented by latitude and longitude coordinates. For these features, we use Hilbert space-filling curves [33] to convert a feature sequence of 2-dimensional values to a feature sequence of 1-dimensional values. It has been found that under most circumstances, linear mappings using Hilbert space-filling curves outperform other mappings [21, 26] in preserving the locality between the elements.

4.1.1 Timeline Attributes

In order to learn the optimal matching ranges for users, timelines are represented by a set of attributes. We consider two sets of attributes which can be categorised as regularity-based and k-gram based attributes as described below.

Regularity Attributes. In this category, we consider attributes aimed at capturing the degree of *regularity* in the timelines. In this regard, we use the regularity statistic, *sample entropy (SampEn)* [27], from the domain of medical data analysis. $SampEn(p, r, n)$ for a time series with n elements is defined as the negative natural logarithm of the conditional probability that two sequences which are similar (within a tolerance, r) for p points remain similar at the next point, where self-matches are not included in calculating the probability [32]. Regularity in the timelines is measured from the perspective of each feature separately. Hence, the $SampEn$ will be computed for individual feature sequence as explained below.

Given $\mathcal{S}_z$, the feature-sequence for the z^{th} feature, p (referred to as epoch length) a positive integer ($p < n$) and r (tolerance) a positive real number, then vectors of length p can be extracted from $\mathcal{S}_z$ as follows: $u_p(i) = \{v_i^z, v_{i+1}^z, ..., v_{i+p-1}^z\}$, where $u_p(i)$ is the i^{th} vector of length p and $1 \leq i \leq n - p + 1$. Let $k_i^p(r)$ be the number of other vectors $u_p(j)$ that are close to the vector $u_p(i)$, i.e., the number of vectors that satisfy $d[u_m(i), u_m(j)] \leq r$, where d is the maximum absolute difference between their scalar components[2], $i \neq j$ and $1 \leq i \leq n - p$ [12, 32]. The sample entropy for feature-sequence $\mathcal{S}_z$ with n elements, for epoch length p and tolerance r, is then defined as:

$$SampEn_z(p, r, n) = -ln \frac{\sum_{i=1}^{n-p} k_i^{p+1}}{\sum_{i=1}^{n-p} k_i^p} \quad (4)$$

Thus, for each user u with timelines of length n and for a fixed tolerance r, we extract the following attributes:

1. $SampEn_z^p$: sample entropy of a feature sequence $\mathcal{S}_z$ for epoch length p,
2. $\mu SampEn_{\mathcal{T}}^p$: mean sample entropy over all feature sequences $\mathcal{S}_z$, $z = 1, 2, ..., m$ of the timeline $\mathcal{T}$ for epoch length p, and
3. $\sigma SampEn_{\mathcal{T}}^p$: standard deviation of sample entropy over all feature sequences $\mathcal{S}_z$, $z = 1, 2, ..., m$ of the timeline $\mathcal{T}$ for epoch length p.

[2] $d[u_p(i), u_p(j)] = max\{|v^z(i+t) - v^z(j+t)| : 0 \leq t \leq p-1\}$.

k-gram Attributes. The second category of attributes which we use to characterise a user's activity patterns are based on k-grams. A k-gram over a feature-sequence $\mathcal{S}_z$ is a k-length ($k > 0$) subsequence of consecutive elements occurring in $\mathcal{S}_z$. k-gram-based attributes have been previously used for sequence classification [41, 15, 39], biological sequence analysis [15] and text classification [8].

For a given k, we extract the following three types of k-gram based attributes from feature-sequences:

1. η_z^k: the total number of distinct k-grams in feature sequence $\mathcal{S}_z$, normalised by the total number of k-grams occurring in $\mathcal{S}_z$,
2. μf_z^k: the mean frequency of occurrence of distinct k-grams in the feature sequence $\mathcal{S}_z$, normalised by the total number of k-grams occurring in $\mathcal{S}_z$, and
3. σf_z^k: the standard deviation of the frequency of occurrence of distinct k-grams in the feature sequence $\mathcal{S}_z$, normalised by the length of $\mathcal{S}_z$.

Note that for $k = 1$, $\eta^1_{mode\ of\ transport}$ represents the total number of distinct modes of transport normalised by the length of timeline.

k-gram-based features are extracted only for symbolic feature sequences; in this work, only feature sequences of *activity names (e.g. modes of transport)* are symbolic. This is because numeric feature sequences, such as *duration* and *start geolocation*, have many distinct k-grams with relatively low frequencies of occurrence for each. As suggested in [24] attributes with low frequencies of occurrence should not be considered for sequence classification. While this problem can be mitigated to some extent by symbolic approximation of numeric sequences, the extraction of k-gram features from such approximations are generally not representative of the actual statistical properties of the sequence.

4.2 Predicting Optimal Matching Unit Ranges

The classification task is to learn the optimal matching range, $\mathcal{N}'$, for each user, where each user is represented by an attribute vector as described above. For this purpose, matching units are divided into three non-overlapping ranges, $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ as shown in Table 1. The optimal matching unit for each range are set as follows: $N'_1 = 1$, $N'_2 = 3$ and $N'_3 = 5$. The rationale for the three ranges selected is to model scenarios in which the next activity performed by individual users depends, to a lesser or greater extent, on their past activity patterns (see Section 5.4 for examples related to the dataset used in this work).

Opt. matching range ($\mathcal{N}_i$)	Opt. matching unit (N'_i)
[0, 1]	1
[2, 4]	3
[5+]	5

Table 1: The three optimal matching ranges $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ used as target classes for classification and the corresponding optimal matching units, N'_1, N'_2 and N'_3 for each class.

5. EVALUATION

We first describe the dataset used to construct activity timelines for users and the experimental methodology employed. This is followed by an evaluation of the proposed N-count based recommender and the classification approach to learn the optimal matching unit range for each user.

5.1 Dataset

Experiments were performed using a subset of GPS trajectory dataset *Geolife Trajectories 1.3*, obtained from the Geolife project [49, 51, 52]. In this dataset, each GPS trajectory in the dataset is a sequence of timestamped points, where each point contains its associated latitude, longitude and altitude. The complete Geolife dataset involves 182 users and their trajectories are distributed over 30 cities in China and few cities of USA and Europe. The trajectories were recorded for a wide range of users' outdoor movements such as going home, work, dining, entertainments, shopping sightseeing and sports activities.

A subset of this trajectory dataset contains labels for the mode of transport associated with the trajectories. For our experiments, this subset is used to build activity timelines for each user. Each activity object in these timelines correspond to an instance of a mode of transport used along with its associated features extracted from the corresponding list of GPS trajectories. This dataset contains 10 different modes of transport, namely, *bike, bus, car, subway, taxi, train, walk, airplane, boat* and *run*. Moreover, each activity object in the timeline contains the following 7 contextual features: *mode of transport, start-time, duration, distance-travelled, average altitude, start* and *end geo-coordinates*. Since the characteristics of the timelines on weekdays and weekends are different, and since less data is available for weekends, here we consider data corresponding to weekdays only. Of this data, a subset of 18 users are selected for the purpose of evaluation; these are the users for which our evaluation methodology allows at least 10 opportunities for recommending the next mode of transport.

Timelines generated from the dataset spanned over 51 days and contained 334 activity objects on average over users. Figure 3 shows the distribution of the total number of modes of transport taken per day by each user, while Figure 4 shows the distribution of the number of distinct modes of transport (divided by total number) taken per day. It can be seen that the median number of the total modes of transport per day for users varies between 3–11 (Figure 3), while the variety of each mode of transport also differs from user to user as shown in Figure 4. These distributions indicate that the timelines generated are reasonably rich with a significant number of modes of transport per day, and that there exists significant variety in the modes of transport taken by different users. For example, the high median value (0.67) in Figure 4 for user 9 shows that this user chooses diverse modes of transport, while the median value for user 1 (0.29) indicates less variety in the modes of transport taken.

Overall, we observe that users take between 2 and 5 distinct modes of transport per day. These figures are inline with expectations given the domain under consideration. Figure 5 shows the percentage of days in which a given mode of transport is taken at least once by users, and indicates that while certain activities are common, others are rare. For example, *walk* and *bus* are popular the modes of transport, while *airplane*, *boat* and *run* rarely occur in the data.

Thus, we conclude that the dataset exhibits significant variation in activity patterns across users and in the modes of transport taken. Moreover, as shown below, our approach leads to good recommendation performance across the user

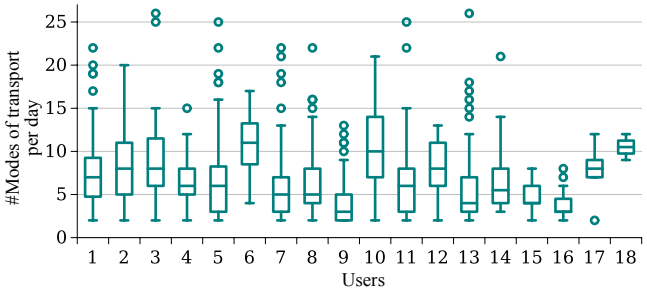

Figure 3: Distribution of the total number of modes of transport per day for each user.

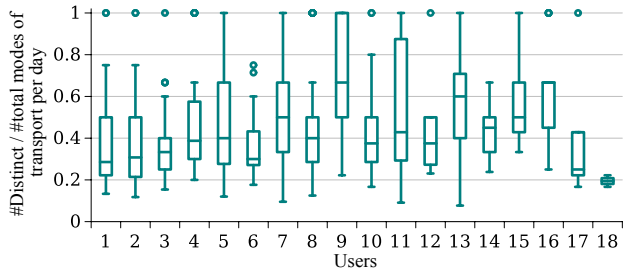

Figure 4: Distribution of the number of distinct modes of transport (divided by total number) per day for each user.

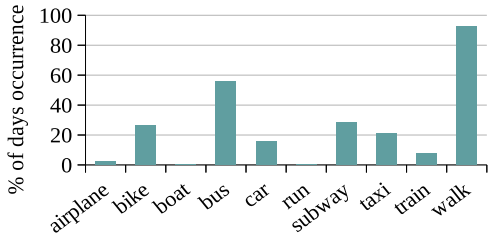

Figure 5: Percentage of days in which each activity occurs at least once over all users.

base and the modes of transport considered, and is not limited to instances with particular characteristics in the data.

5.2 Methodology

5.2.1 Recommendation Algorithm

An offline evaluation was conducted for the proposed recommendation approach. Each user's complete timeline was split into training and test timelines, where the test timeline contained data for the most recent 20% of available days. For each user, a ranked list of recommended modes of transport were generated at different *recommendation times* (RT), which corresponded to the end time of each activity object in the test timeline. In each case, the target of recommendation is the next mode of transport in the timeline. Recommendation accuracy for each RT is computed as the reciprocal of the rank position of the target mode of transport in the recommended list. The mean reciprocal rank (MRR) is then computed over the RTs for each user. Furthermore, in the N-count timeline matching approach, the matching unit is varied (between $N = 0-10$) to identify the observed optimal matching unit for each user.

For the computation of two-level edit distances between

$SampEn^p_{transport-mode}$	$SampEn^p_{start-time}$
$SampEn^p_{duration}$	$SampEn^p_{distance-travelled}$
$SampEn^p_{start-geo}$	$SampEn^p_{end-geo}$
$SampEn^p_{avg-altitude}$	
$\mu SampEn^p$	$\sigma SampEn^p$
$\eta^1_{transport-mode}$	$\eta^k_{transport-mode}$
$\mu f^k_{transport-mode}$	$\sigma f^k_{transport-mode}$

Table 2: List of attributes extracted from user timelines. Here, $p = 2, 3$ and $k = 2, 3$.

timelines, the following operation costs and feature weights were used: $c_{ins} = c_{del} = 1$, and $c_{sub} = 2$; $w_{mode-of-transport} = 3$, $w_{start-time} = 1$, $w_{duration} = 0.5$, $w_{distance-travelled} = 3$, $w_{start-geocoordinates} = 0.3$, $w_{start-geocoordinates} = 0.3$, $w_{avg-altitude} = 0.2$. (See [23] for details on the two-level edit distance approach.) In the above, the weight associated with updating the mode of transport (i.e. the activity name such as *bus*, *walk*, *taxi*, etc.) was set to the highest value since this is clearly a key consideration when computing distances between timelines. The value of this weight, along with those for start time, duration, start/end geolocation and average-altitude, are set according to their hypothesised importance from the perspective of comparing timelines.

5.2.2 Learning Optimal Matching Unit Range

The classifier for learning the optimal matching unit range is evaluated using a leave-one-out cross-validation. Since each user is represented by an attribute vector extracted from its timeline, the number of instances is same as the number of users. Table 2 lists the attributes used to construct the attribute vectors for user timelines. For the computation of sample entropy ($SampEn$), values of $p = 2, 3$ are used, since our experiments showed that $p > 3$ did not improve the classification results. Similarly, for the k-gram based attributes, values of $k = 2, 3$ are also used. The tolerance value, r, is set to 0 for the categorical mode of transport feature sequence [3], while for numeric feature sequences such as duration, average altitude, etc., $r = 0.15 \times \sigma$, where σ is the standard deviation of the values in the sequence.

The ground truth for the instances are the three ranges of matching units, i.e., $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ determined from the observed optimal matching units over all users (see Table 1 for the ranges considered). For each user, a subset of the classification attributes are selected using a wrapper approach [22], using the C4.5 decision tree induction algorithm [30], greedy backward search and area under the ROC curve as the evaluation measure. We use a leave-one-out approach to generate 18 different training sets for each of the 18 users, and an internal 10-fold cross validation is performed on each training set. For simplicity, the same subset of attributes is selected for all users, i.e. the most common subset which was found for 12 (of the 18) users in the dataset.

The pruned attribute vectors for each user are then fed into a C4.5 induction algorithm, which is evaluated using

[3]Sample entropy is traditionally applied to numeric sequences [27]. Here, we also apply sample entropy to symbolic sequences (i.e. feature sequences of modes of transport) by assigning a numeric value to each distinct mode of transport and using a zero tolerance, $r = 0$. This makes sample entropy independent of the scale and ordinality of the numeric values assigned to the symbols, thus preserving the semantics of sample entropy in the case of symbolic sequences.

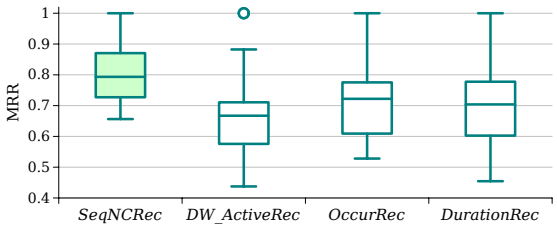

Figure 6: MRR distribution over all users for the *SeqNCRec*, *DW_ActivRec*, *OccurRec* and *DurationRec* recommenders.

leave-one-out cross validation. The output of this classification gives the matching unit range for each user.

5.3 Recommendation Performance

In order to evaluate the performance of our sequence-based N-count recommendation algorithm (*SeqNCRec*), the following baseline approaches are considered:

- The *daywise sequence-based* recommender (*DW_ActivRec*) is the algorithm proposed in our earlier work [23]. This approach uses the same two-level distance metric to compare timelines, however it uses a *daywise* matching of timelines as described in Section 3.3.2.
- The *high occurrence* recommender (*OccurRec*) assumes that the best modes of transport are those which are taken most often. Thus for each user, it recommends modes of transport in decreasing order of how often they are taken by that user in the training set.
- The *high duration* recommender (*DurationRec*) assumes that the best modes of transport are those in which users spend most of their time. Thus for each user, it recommends modes of transport in decreasing order of the time spent in them by that user in the training set.

For this comparison, the MRR values for *SeqNCRec* are obtained using the observed optimal matching units for each user. Figure 6 shows the distribution of MRR over all users using *SeqNCRec* and the three baseline approaches described above. The results show that *SeqNCRec* approach outperforms the other three baselines for the majority of users. The median MRR is 19%, 10% and 13% higher when using *SeqNCRec* compared to the baseline *DW_ActivRec, OccurRec* and *DurationRec* approaches, respectively.

Figure 7 shows the MRR values for each user individually using the proposed *SeqNCRec* algorithm. It is clear that good MRR values are obtained for all users with a high mean (0.81) over all the users. Moreover, comparing these results with the median number of distinct activities per day in Figure 4, it can be seen that the recommendation algorithm also performs well for those users which have a high degree of variety in their transport patterns, which is clearly an important finding. For example, the MRR for user 9 exceeds that for user 12, although user 9 has a much greater variety in the modes of transport taken in comparison to user 12.

We also compare performance of the *SeqNCRec* recommender across the different modes of transport, i.e. when each mode of transport represented the target activity for recommendation. Figure 8 shows the distribution of mean reciprocal ranks grouped by the target mode of transport. Comparing these results with Figure 5, which shows the popularity of transport modes, it can be seen that the recommendation algorithm is not limited to making just the "obvious" recommendations. For example, the recommendation performance is similar on average for *bus*, *bike* and *car* (all have the same median MRR), even though *bus* is a more popular mode of transport compared to *bike* and *car*.

5.4 Classification Performance

In the above, results for the *SeqNCRec* recommender were obtained using the observed optimal matching unit for each user; i.e. the value of N (using N-count matching) which achieved the best MRR for each user was selected. Here, we consider the classification approach as described in Section 4 to learn the optimal matching unit for each user.

We begin by motivating the choice of matching unit ranges used in this work. Figure 9 shows the observed optimal MRR for three representative users against matching unit using the *SeqNCRec* algorithm. The results show different trends for each of these users. For example, the greatest MRR value is achieved for user 12 at a matching unit of 0 (i.e., using only the current activity object), which decreases as the matching unit increases. In contrast, MRR for user 5 peaks at a matching unit of 2 and declines thereafter, while the MRR for user 4 peaks later at matching unit equal to 5. As different users will have different activity patterns, it is not surprising that their MRR varies with matching unit. As such, these results confirm the need to learn optimal matching units for each user. However, since users will naturally exhibit variation in their activity patterns, here we adopt the approach of learning optimal matching unit ranges, and choosing a fixed matching unit value from within each range.

We now turn to the performance achieved by the matching unit range classifier. Using a ground truth based on observed optimal matching units, the number of instances (users) in each of the three classes $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ were 8, 6 and 4, respectively. Using the wrapper approach for attribute subset selection, attributes $SampEn^2_{duration}$, $SampEn^2_{avg-altitude}$ and $\mu SampEn^3$ were selected to build the decision trees for each user. Figure 10 shows the results using a leave-one-out cross-validation approach. As can be seen, good classification performance is achieved across all three classes, with weighted precision and recall both equal to 0.78.

While the above classification results are promising, the main test of this approach is whether the matching units N'_i associated with each predicted range $\mathcal{N}'_i$ lead to high quality recommendations for users. The results show that generally only small differences in MRR are seen; the mean reduction in MRR is 3.1%, with a standard deviation of 5.3%. Thus, we can conclude that our classification approach learns personalised matching units for users (in the domain considered) sufficiently well from a recommendation perspective.

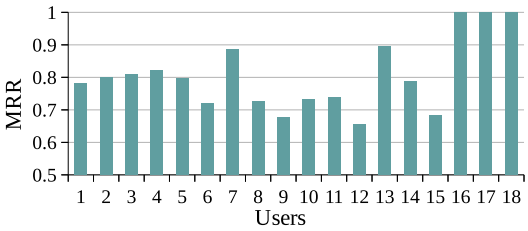

Figure 7: MRR achieved for each user by the *SeqNCRec* recommender using observed optimal matching units.

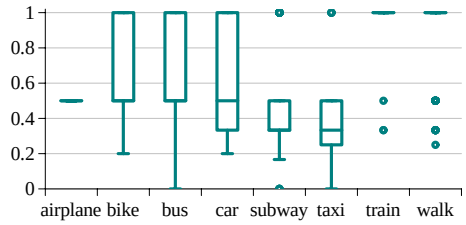

Figure 8: Distribution of MRR for each target activity across all users using the $SeqNCRec$ recommender.

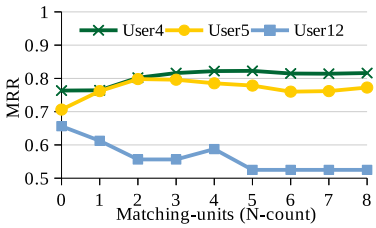

Figure 9: MRR versus matching unit (N-count) for three representative users.

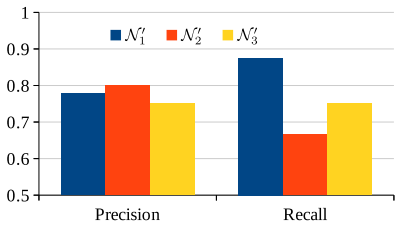

Figure 10: Precision/recall for target classes $\mathcal{N}_1'$, $\mathcal{N}_2'$ and $\mathcal{N}_3'$.

6. CONCLUSIONS AND FUTURE WORK

In this work we have proposed a content-based approach to recommend the next mode of transport to users in which sequence-based modeling is used to capture the order as well as the context associated with user activities. We introduced a timeline matching approach for generating these sequence-based recommendations. A classification approach to learn the personalised optimal matching units for users was also proposed. Evaluations using a real-world dataset show good results from a recommendation perspective and for the matching unit learning approach.

There is rich scope for future work in this area. For example, the current work does not take into account socio-economic characteristics and user demographics, and travel variables such as car ownership, bus/train availability and other urban settings, which are also important [10] for transport choice. Moreover, the issue of scalability when dealing with large-sized datasets is an important consideration; in this regard, one possible solution is to limit (based on recency, day, or time) the number of candidate timelines used when making recommendations. Our approach also lends itself to more sophisticated recommendation scenarios — instead of simply suggesting the next activity to users, a sequence of activities, along with associated contextual data, can be recommended (for example, suggesting a sequence of tourist attractions to users, and when and with whom they can be visited). Finally, a hybrid approach to recommendation can also be considered by, for example, applying a collaborative filtering algorithm in conjunction with the current content-based approach, which can alleviate sparsity issues [42] and further improve recommendation quality.

7. ACKNOWLEDGMENTS

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

References

[1] Microsoft Cortana. http://windows.microsoft.com/en-us/windows-10/getstarted-what-is-cortana.

[2] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3), 2011.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[4] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: A survey. *Geoinformatica*, 19(3):525–565, July 2015.

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Know.-Based Syst.*, 46:109–132, July 2013.

[6] T. Bohnenberger and A. Jameson. When policies are better than plans: Decision-theoretic planning of recommendation sequences. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, pages 21–24, New York, NY, USA, 2001. ACM.

[7] G. Bonnin, A. Brun, and A. Boyer. Using skipping for sequence-based collaborative filtering. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1 of *WI-IAT '08*, pages 775–779, Washington, DC, USA, 2008. IEEE Computer Society.

[8] W. B. Cavnar and J. M. Trenkle. N-Gram-Based Text Categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US, 1994.

[9] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 714–722, New York, NY, USA, 2012. ACM.

[10] N. Commins and A. Nolan. The determinants of mode of transport to work in the greater dublin area. *Transport Policy*, 18(1):259 – 268, 2011.

[11] M. Costa, A. L. Goldberger, and C.-K. Peng. Multiscale entropy analysis of complex physiologic time series. *Physical review letters*, 89(6):068102, 2002.

[12] M. Costa, A. L. Goldberger, and C.-K. Peng. Multiscale entropy analysis of biological signals. *Physical review E*, 71(2):021906, 2005.

[13] M. D. Costa and A. L. Goldberger. Generalized multiscale entropy analysis: Application to quantifying the complex volatility of human heartbeat time series. *Entropy*, 17(3):1197–1203, 2015.

[14] M. Deshpande and G. Karypis. Selective Markov models for predicting web page accesses. *ACM Trans. Internet Technol.*, 4(2):163–184, May 2004.

[15] G. Dong and J. Pei. *Sequence Data Mining (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[16] J.-P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of modern physics*, 57(3):617, 1985.

[17] L. Gong, T. Morikawa, T. Yamamoto, and H. Sato. Deriving personal trip data from GPS data: A literature review on the existing methodologies. *Procedia - Social and Behavioral Sciences*, 138:557 – 565, 2014. The 9th International Conference on Traffic and Transportation Studies (ICTTS 2014).

[18] P. Grassberger and I. Procaccia. Estimation of the Kolmogorov entropy from a chaotic signal. *Phys. Rev. A*, 28:2591–2593, Oct 1983.

[19] R. Guha, V. Gupta, V. Raghunathan, and R. Srikant. User modeling for a personal assistant. In *Proceedings of the Eighth ACM International Conference on Web*

Search and Data Mining, WSDM '15, pages 275–284, New York, NY, USA, 2015. ACM.

[20] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 131–138, New York, NY, USA, 2012. ACM.

[21] H. V. Jagadish. Linear clustering of objects with multiple attributes. *SIGMOD Rec.*, 19(2):332–342, May 1990.

[22] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, Dec. 1997.

[23] G. Kumar, H. Jerbi, C. Gurrin, and M. P. O'Mahony. Towards activity recommendation from lifelogs. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '14, pages 87–96, New York, NY, USA, 2014. ACM.

[24] N. Lesh, M. J. Zaki, and M. Ogihara. Mining features for sequence classification. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 342–346, New York, NY, USA, 1999. ACM.

[25] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 34:1–34:10, New York, NY, USA, 2008. ACM.

[26] B. Moon, H. Jagadish, C. Faloutsos, and J. Saltz. Analysis of the clustering properties of the Hilbert space-filling curve. *Knowledge and Data Engineering, IEEE Transactions on*, 13(1):124–141, Jan 2001.

[27] S. Pincus, I. Gladstone, and R. Ehrenkranz. A regularity statistic for medical data analysis. *Journal of Clinical Monitoring*, 7(4):335–345, 1991.

[28] S. M. Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[29] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of the 2nd Conference on USENIX Symposium on Internet Technologies and Systems - Volume 2*, USITS'99, pages 13–13, Berkeley, CA, USA, 1999. USENIX Association.

[30] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[31] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[32] J. S. Richman and J. R. Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[33] H. Sagan. *Space-filling curves*. Springer Science & Business Media, 2012.

[34] M. A. Shafique and E. Hato. Use of acceleration data for transportation mode prediction. *Transportation*, 42(1):163–188, 2015.

[35] G. Shani, D. Heckerman, and R. I. Brafman. An MDP-based recommender system. *J. Mach. Learn. Res.*, 6:1265–1295, Dec. 2005.

[36] M. O. Sokunbi. Sample entropy reveals high discriminative power between young and elderly adults in short fMRI data sets. *Frontiers in neuroinformatics*, 8, 2014.

[37] Y. Sun, N. J. Yuan, X. Xie, K. McDonald, and R. Zhang. Collaborative nowcasting for contextual recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 1407–1418, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[38] P. Symeonidis, A. Papadimitriou, Y. Manolopoulos, P. Senkul, and I. Toroslu. Geo-social recommendations based on incremental tensor reduction and local path traversal. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, pages 89–96, New York, NY, USA, 2011. ACM.

[39] A. Tomović, P. Janičić, and V. Kešelj. n-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer methods and programs in biomedicine*, 81(2):137–153, 2006.

[40] C.-Y. Wang, Y.-H. Wu, and S.-C. T. Chou. Toward a ubiquitous personalized daily-life activity recommendation service with contextual information: A services science perspective. *Information Systems and E-Business Management*, 8(1):13–32, January 2010.

[41] Z. Xing, J. Pei, and E. Keogh. A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, 12(1):40–48, Nov. 2010.

[42] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, Jan 2015.

[43] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Smart itinerary recommendation based on user-generated GPS trajectories. In *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing*, UIC'10, pages 19–34, Berlin, Heidelberg, 2010. Springer-Verlag.

[44] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI 2010*. Association for Computing Machinery, Inc., July 2010.

[45] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1029–1038, New York, NY, USA, 2010. ACM.

[46] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. *Artif. Intell.*, 184-185:17–37, June 2012.

[47] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55, Sept. 2014.

[48] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. Understanding transportation modes based on GPS data for web applications. *ACM Trans. Web*, 4(1):1:1–1:36, Jan. 2010.

[49] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 312–321, New York, NY, USA, 2008. ACM.

[50] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw GPS data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 247–256, New York, NY, USA, 2008. ACM.

[51] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Database Engineering Bulletin*, June 2010.

[52] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA, 2009. ACM.

[53] Y. Zheng and X. Zhou. *Computing with Spatial Trajectories*. Springer Publishing Company, Incorporated, 2014.