On Learning Prediction Models for Tourists Paths

Cristina Ioana Muntean, ISTI–CNR
Franco Maria Nardini, ISTI–CNR
Fabrizio Silvestri, Yahoo! Research Labs
Ranieri Baraglia, ISTI–CNR

In this paper, we tackle the problem of predicting the "next" geographical position of a tourist, given her history (i.e., the prediction is done accordingly to the tourist's current trail) by means of supervised learning techniques, namely Gradient Boosted Regression Trees and Ranking SVM. The learning is done on the basis of an object space represented by a 68 dimension feature vector, specifically designed for tourism related data. Furthermore, we propose a thorough comparison of several methods that are considered state-of-the-art in recommender and trail prediction systems for tourism, as well as a popularity baseline. Experiments show that the methods we propose consistently outperform the baselines and provide strong evidence of the performance and robustness of our solutions.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Geographical PoI Prediction, Learning to Rank

ACM Reference Format:
Cristina Ioana Muntean, Franco Maria Nardini, Fabrizio Silvestri, Ranieri Baraglia 2015. On Learning Prediction Models for Tourists Paths. *ACM Trans. Intell. Syst. Technol.* V, N, Article A (January 2015), 35 pages.
DOI: http://dx.doi.org/10.1145/2766459

1. INTRODUCTION

Tourism applications spark an increasing interest from industry and academia due to the continous growth of user generated content from multiple geo-referenced sources. The aim is helping tourists make smarter decisions in different scenarios involving recommendation, intelligent planning, booking, etc. Such systems are used within location-based social networks and targeted mobile applications.

This work extends LEARNEXT [Baraglia et al. 2013], a "next-tourist-place" predictor allowing the provisioning of the "next" most likely place that a tourist will visit in a city. The approach we propose can be used as a building block for more complex applications, such as devising suggestions regarding places of interest when visiting a city and making effective predictions of the tourist behavior in a city.

LEARNEXT predicts tourist places according to the footprints of a tourist while visiting a city and a history of previously visited places (i.e., *visit patterns*) by other tourists.

This work was partially supported by the EU projects InGeoCLOUDS (no. 297300), MIDAS (no. 318786), E-CLOUD (no. 325091), the Italian PRIN 2011 project "Algoritmica delle Reti Sociali Tecno-Mediate" (2013-2014) and the Regional (Tuscany) project SECURE! (FESR PorCreo 2007-2011). Author's addresses: C.I. Muntean (Corresponding Author), F.M. Nardini, R. Baraglia, High Performance Computing Laboratory, ISTI–CNR, via G. Moruzzi, 1, 56124, Pisa, Italy. F. Silvestri, Yahoo Labs, London, UK.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2015 ACM. 2157-6904/2015/01-ARTA $15.00
DOI: http://dx.doi.org/10.1145/2766459

For the selection of tourist sites, the system uses a set of *Points of Interest* (PoIs) identified a priori.

The work published in [Baraglia et al. 2013] presents the preliminary results for the task of predicting the "next" PoI for tourism. While our intuition showed the proposed approach works, further investigation is required in understanding why the method has such promising results and what are the factors contributing to that. This version of the work wishes to answer four new research questions. We do more than predicting where a tourist would move next, in fact we want to understand the signals that interfere with the tourists movements in cities of different dimensions by varying the conditions in which they move. For example, the previous analysis of LEARNEXT does not explain how the performance varies by changing the set of features used or how the learning parameters affect the prediction power of the methods.

In order to address these challenging tasks and more, in the following we present a deeper analysis of the performance of LEARNEXT. In particular, we answer the following new research questions:

(1) What is the robustness of the proposed techniques? In particular, how does the performance change by varying the learning parameters? We answer this question by varying the learning parameters and evaluating each configuration for the two ML techniques used in our framework.
(2) How does the trail length affect the effectiveness of the proposed models? Are the proposed methods able to deal with trails of different lengths? If yes, how does this property affect the effectiveness of the prediction? We answer this question by analysing the prediction accuracy when employing trail sets of different lengths in building the models.
(3) How does the popularity of PoIs affect the prediction? Is predicting a popular PoI easier than predicting a rare PoI? We answer this question by performing an analysis of the effectiveness of our proposed techniques on two distinct classes of PoIs (frequent and rare).
(4) What is the sensitivity of the model w.r.t. the proposed features? How does the performance of the model change by varying the set of features employed? We perform an analysis to understand what are the features that contribute most to the effectiveness of the prediction, including an indepth analysis of the top 5 most discriminant ones. Are those features always the same for each city? We answer these questions using three different approaches, one of which a graph-based approach for selecting significant features based on the correlation between pairs of features, and several methods for assessing the importance of the features employed in a prediction model.

LEARNEXT is structured in two modules: one operating offline and one operating online w.r.t. the current visit of a tourist. The offline module is used to build the knowledge model that is in turn used for predicting tourist behavior. The online module uses information from the current visit of a tourist and the knowledge model to predict the next location. Recommending next PoIs is quite a challenging task. One would expect that suggesting the most frequently visited set of PoIs would provide high quality recommendations. In fact, as show in Section 4, such a baseline performs quite poorly compared to the methods we proposed.

The paper is structured as follows: in Section 2 we present the related work regarding tourism recommendations and prediction. In Section 3 we define the problem of predicting the next tag as a structured learning problem, whereas Section 3.1 details the machine learning methods we adopt to solve the problem and Section 3.2 details the features we consider for modeling PoIs and tourist trails. Section 4 describes the dataset and the methodology we use to assess our solutions together with a thorough evaluation of the effectiveness of our methods. Finally, Section 5 concludes the current research and draws up future work directions.

2. RELATED WORK

This paper explores the problem of predicting the most likely "*Point of Interest*" (PoI) to be visited by a tourist during her tour in a city. It involves two appealing fields of research in the tourism scenario: *data analysis* and *PoI prediction/recommendation*. The first focuses on the analysis of the photo traces left by tourists when visiting a city and the second studies techniques to predict/recommend interesting PoIs exploiting knowledge mined from historical data.

2.1. Data Analysis.

A significant number of papers relies on mining geo-spatial and textual metadata associated with Flickr images. Important efforts have been spent in analyzing the dynamics of people moving through cities [Girardin 2009]. Girardin *et al.* study explicit (e.g. Flickr photos) or implicit (e.g. cell association in a mobile communication network) digital footprints that people leave behind while traveling through a city [Girardin et al. 2008a]. Girardin *et al.* [Girardin et al. 2008b] tap tourist dynamics for better urban planning and deployment of location-based services. Moreover, Rattenbury *et al.* [Rattenbury et al. 2007] analyze the geo-temporal dynamics of Flickr tags in order to distinguish between tags describing places and events. Popescu and Grefenstette [Popescu and Grefenstette 2009] deduce visit times at landmarks based on timestamps of Flickr photos, while Ahern *et al.* [Ahern et al. 2007] plot aggregated textual metadata associated with geo-referenced Flickr images on a map interface with the aim of exposing how Flickr users at large describe landmarks. Crandall *et al.* study the association of Flickr photos to physical locations [Crandall et al. 2009]. They also apply their techniques to extract landmarks corresponding to a geo-spatial hierarchy. Arase *et al.* [Arase et al. 2010] apply pattern mining to 5.7 million of geo-tagged photos to perform photo trip pattern mining. Experimental result shows that the proposed method outperforms the baselines in segmenting photo collections into photo trips.

2.2. PoI Prediction/Recommendation.

A first approach to solve the PoI prediction problem uses trajectory pattern mining to devise temporally-annotated common patterns (trajectories) of movements from data. Trajectories are a concise representation of the behavior of moving objects as sequences of regions frequently visited with typical travel time. Trajectory-based models are exploited in [Monreale et al. 2009], [Baraglia et al. 2012], [Krumm and Horvitz 2006] to predict the most likely locations that are of interest for a tourist.

Monreale *et al.* propose "WhereNext", a method predicting the next location of a moving object [Monreale et al. 2009]. A decision tree, named T-pattern Tree, is built and evaluated with a formal training and test process. The tree is learned from the trajectory patterns within a certain area, and it is used as a predictor for the next location of a new trajectory by finding, on the tree, the best matching path. Authors propose three different matching methods to classify a new moving object. Furthermore, they study their impact on the quality of the prediction. Finally, the authors show an exhaustive set of experiments and results on real-world datasets. Moreover, a set of different measures aimed at evaluating a-priori the predictive power of a set of trajectory patterns has been proposed and tuned on a real life case study. Since this contribution, by using the trajectory pattern matching approach, is designed for predicting the next PoI for a given trajectory, we will use it as a baseline in the experimental evaluation of our proposed method, in Section 4.

Krumm and Horvitz propose a trajectory-based system, called Predestination [Krumm and Horvitz 2006]. It tries to predict the location of a certain vehicle as a natural progression of a trip by exploiting previous covered trajectories and other information, such as the geological nature of the territory and the driver history. The system tries to directly calculate the point a driver will find after a certain period of time. Multiple analysis components are fused together using Bayesian inference to produce a map of the probabilistic destinations. The space is divided into a grid of square cells, and each cell can be seen as a destination. The purpose of Predestination is to predict in which cell a trip will end by studying the cells that a traveler has already traversed and the characteristics of each one of them. It exploits a method that computes the probability of each cell to be the final destination. Many random variables such as the geological nature of the territory of a cell are used by the adopted model. Authors test the system on a set of GPS trajectories obtained from 169 subjects over a total of over $7,000$ tracks.

Similar efforts have been spent in solving the PoI recommendation task. Here, the problem deals with generating a list of possible PoIs that are of interest for a tourist. It differs from the prediction task as it aims at maximizing the satisfaction of a user during her tour of the city, while the first one aims at identifying only one PoI as the first candidate to be visited. In [Levandoski et al. 2012], a location-aware recommender system (LARS), that uses location-based ratings to produce recommendations, is proposed. LARS exploits user ratings of locations through user partitioning, a technique that influences recommendations with ratings spatially close to querying users. It maximizes system scalability while not sacrificing recommendation quality. Experiments conducted on large-scale real-world data from Foursquare and MovieLens reveal that LARS is efficient and capable of producing recommendations twice as accurate compared with state-of-the-art competitors.

Ye *et al.* [Ye et al. 2010], [Ye et al. 2011] realize location recommendation services for large-scale location-based social networks by exploiting the social and geographical characteristics of users and locations/places. Authors develop a friend-based collaborative filtering approach for location recommendation based on collaborative ratings of places made by social friends. They also propose a unified PoI recommendation framework, which fuses user preference to a PoI with social and geographical influences. Results show that the unified collaborative recommendation approach significantly outperforms a wide spectrum of alternative recommendation approaches.

Zheng *et al.* perform travel recommendations by mining multiple users' GPS traces [Zheng and Xing 2011], [Zheng et al. 2009]. They model multiple users' location histories with a tree-based hierarchical graph. Based on the graph, authors propose a HITS-based inference model, which regards the access of an individual in a location as a directed link from the user to that location. The recommendation process uses a collaborative filtering model that infers a user's interests in an unvisited location based on her location histories together with those of others tourists. Results show that the HITS-based inference model outperforms baseline approaches like rank-by-count and rank-by-frequency.

Noulas *et al.* [Noulas et al. 2012] study the problem of predicting the next venue a mobile user will visit (in Foursquare-like terminology, the next *check-in*) by exploring the predictive power offered by different aspects of the user behavior. Authors propose a set of 12 features that aims to capture the factors that may drive users' movements. They model transitions between types of places, mobility flows between venues, and spatio-temporal characteristics of user check-in patterns. Furthermore, they exploit such features in two supervised learning models, based on linear regression and M5 model trees, resulting in a higher overall prediction accuracy.

They model the task as a binary classification problem, whereas we model it as a next PoI ranking problem, based on the likelihood of each PoI to be next in the user trail. It is worth noting that the dataset used in this approach is different from ours in both type of movements and places: ours is focused on tourist movements within a city, while both Foursquare and Facebook Places data contains check-in such as workplaces, homes, restaurants, meeting places with friends, etc., and not necessarily points of tourist interest. The goal of tourists is to visit as many places in a limited time span, leading to longer trails whereas on Foursquare data is more oriented towards recording movements of regular users in a day, tending to shorter and more repetitive trajectories. Moreover, we propose a broader set of features, originated from a Flickr dataset, capturing more dimensions of the tourist behavior. We cast the prediction problem into a *learning to rank* task which allows us to use two effective machine-learned techniques (Ranking SVM [Joachims 2006] and GBRT [Zheng et al. 2007]) to solve it.

Another important contribution by Lian *et al.* studies the limit of check-in location predictability, i.e., to what extent the next locations are predictable, in the presence of special properties of check-in traces [Lian et al. 2014]. Authors starts by estimating the entropy of an individual check-in trace and then leverage Fano's inequality to transform it to predictability. Authors propose an extensive analysis on two large-scale check-in datasets from Jiepang and Gowalla with 36M and 6M check-ins, respectively. As a result, they find 25% and 38% potential predictability respectively. Finally, a correlation analysis between predictability and users' demographics has been performed. The results show that the demographics, such as gender and age, are significantly correlated with location predictability.

PoI prediction has been also investigated in [Sadilek et al. 2012]. Here, authors present FLAP, a system that solves two closely related tasks: *link* and *location* prediction in online social networks. For link predictions, FLAP infers social ties by considering patterns in friendship formation, the content of people's messages, and user location. Authors show that while each component is a weak predictor of friendship alone, combining them results in a strong model, accurately identifying the majority of friendships. For location prediction, FLAP implements a scalable probabilistic model of human mobility, where each user with known GPS positions becomes a noisy sensors of the location of their friends. Authors propose supervised and unsupervised learning scenarios. Moreover, they evaluate FLAP on a large sample of highly active users from two distinct geographical areas. Authors show that i) it reconstructs the entire friendship graph with high accuracy even when no edges are given; and ii) it infers users' fine-grained location, even when they keep their data private and we can only access the location of their friends.

We are investigating techniques for predicting the "next" PoI visited by a tourist. The approaches above are thus different from ours because authors of [Lian et al. 2014; Sadilek et al. 2012] do not focus on this particular class of people, namely tourists, moving in a geographical area. Our approach is different from the one above because we do not take into account social network information for computing the prediction. We only exploit sequences of previously visited PoIs of a tourist for computing "next" likely locations that will be visited in the future.

Another recent work exploiting social and historical relations to model users' check-in prediction is [Gao et al. 2012]. Here, Gao *et al.* propose a social-historical model to explore user's check-in behavior on Location-Based Social Networks (LBSNs). Authors employ a model that integrates the social and historical effects and assesses the role of social correlation in user check-in behavior. In particular, the model captures the property of user check-in history in forms of power-law distribution and short-term effect, and helps in explaining the check-in behavior. The experimental results on a real world LBSN demonstrate that the proposed approach properly models user check-ins and shows how social and historical ties can help predicting locations.

Cho *et al.* also propose a characterization of human movements and their correlation with social networks [Cho et al. 2011]. In particular, authors aim to understand what basic laws govern human motion and dynamics. They find that users experience a combination of: i) periodic movements that are geographically limited and 2) occasionally, movements that seem random jumps. Short-ranged travel is periodic both spatially and temporally and not effected by the social network structure, while long-distance travel is more influenced by social network ties. Based on the above findings, authors develop a model of human mobility that combines periodic short range movements with travel due to the social network structure.

A similar social-oriented approach to ours exploits a dataset from Facebook Places to understand the factors that influence where users check in, including previous check-ins, similarity to other places, where their friends check in, time of day, and demographics [Chang and Sun 2011]. Authors show how users respond to their friends' check-ins and which factors contribute to users liking or commenting on them. They also show how this can be used to improve the ranking of check-in stories, ensuring that users see only the most relevant updates from their friends. Finally, authors construct a model to predict friendship based on check-in count and show that co-check-ins has a statistically significant effect on friendship.

Lian *et al.* [Lian et al. 2015] propose a Collaborative Exploration and Periodically Returning model (CEPR), based on a novel problem, Exploration Prediction (EP), which forecasts whether people will seek unvisited locations to visit. The evaluation results show that EP achieves a roughly 20% classification error rate outperforming the baselines. Moreover, results confirm that CEPR improves performance up to 30% compared to the traditional location prediction algorithms.

Lucchese *et al.* propose an algorithm which interactively generates personalized recommendations of places based on the knowledge mined from photo albums and Wikipedia [Lucchese et al. 2012]. The authors introduce the model as a graph-based representation of the knowledge, and exploits random walks with restart to select the most relevant PoIs for a specific user. The basic idea is that the recommender system relies on an initial set of PoIs to be used as query places. Query places are important because they represent contextual information identifying tourists sites. The proposed recommendation algorithm has been evaluated by checking whether the suggested PoIs were actually posted in albums on Flickr. The approach by Lucchese *et al.* is particularly tailored for tourism. It also exploits photos from Flickr and data from Wikipedia for devising the recommendation. Moreover, results show it is an effective competitor. For these reasons, in Section 4 we presents a comprehensive comparison of this approach against our proposal.

3. THE LEARNEXT PROBLEM

Let $P = \{p_1, \ldots, p_N\}$ be the set of PoIs in a given tourist location. Let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of tourists that visited the location in the past. We assume a tourist $u_i \in U$ has visited a subset of the PoIs, $V_i \subseteq P$, $V_i = \{v_1, v_2, \ldots, v_k\}$. Moreover, we define $Z_i = P \setminus V_i$ as the subset of PoIs not yet visited by u_i.

Definition 3.1 (*Trail*). Given a tourist u_i and her set of visited PoIs V_i, we define as trail $\hat{V}_i$ a temporally ordered sequence of k PoIs in V_i.

Let $\hat{Y} = \langle y_1, y_2, \ldots, y_{|Z_i|} \rangle$ be the ordering (i.e., a permutation) for the PoIs in Z_i, such that y_1 is the PoI that a tourist will likely visit after v_k, y_2 the second one, etc. Finally, let Y be a general permutation of PoIs in Z_i.

We define the LEARNEXT problem as follows. The goal is to learn a ranking function γ over the class of ranking functions H, such that a given loss function $\Delta\left(Y_{\gamma}, \hat{Y}\right)$ is minimized. The loss function measures the penalty of having ordered PoIs in Z_i as in Y instead of having outputted the correct ordering $\hat{Y}$.

Given a training set $S \subseteq 2^P$, a subset of 2^P - the universe of all permutations of PoIs in P, we aim at minimizing a given loss function $\Delta\left(Y_f, \hat{Y}\right)$, over all the ranking functions $f \in H$,

$$\gamma = \operatorname*{argmin}_{f \in H} \sum_{S} \Delta\left(Y_f, \hat{Y}\right) \tag{1}$$

We thus aim at solving the LEARNEXT problem by learning from a given training set of data the best function γ that can predict the ranking of PoIs, which a tourist has not yet visited, according to the probability of being the next in the trail.

3.1. Machine Learning Based Models

The above LEARNEXT problem can be seen as a *learning to rank* problem [Liu 2009]. *Learning to rank* allows to build models able to order PoIs following their decreasing likelihood of being visited as the next PoI by a given tourist.

Machine learning (ML), in fact, allows to learn from data the function γ that minimizes the error of a given loss function $\Delta\left(Y, \hat{Y}\right)$. This way, the LEARNEXT problem becomes a supervised ML problem solved by building a model which ranks highest the PoI with the highest likelihood of being visited next by a tourist. Models are trained on datasets containing examples built from trails of tourists moving in a city. Each PoI is represented by a high-dimensional feature vector and its associated label indicates the PoI's likelihood of being visited next for the particular tourist u_i within her current trail $\hat{V}_i$.

We build our ranking models by relying on two well-known *learning to rank* techniques: Ranking SVM [Joachims 2002, 2006] and Gradient Boosted Regression Trees (GBRT) [Zheng et al. 2007]. Ranking SVM is a "pairwise" *learning to rank* technique based on the well-known Support Vector Machines. The method aims at learning a retrieval function that maximizes the empirical Kendall's τ [Baeza-Yates et al. 1999] on the training set, i.e., $\tau(Y_f, \hat{Y}))$. Joachims prove that maximizing Kendall's τ is equivalent to minimize the "number of mis-ordered pairs in the ordering" loss function during learning [Joachims 2002]. In this context, the "number of mis-ordered pairs in the ordering" loss function can be written as the "number of mis-ordered pairs of PoIs in a given Y_f". We do not detail here the mathematical formulation proving the relations between Kendall's τ and such loss function. Interested readers can find it in [Joachims 2002].

Gradient Boosted Regression Trees (GBRT) works by building an ensemble of regression trees, typically of limited depth. At each iteration the algorithm builds a new tree with the aim of minimizing the root mean squared error (RMSE). RMSE is a "pointwise" loss function. While training the model, it measures the error between the prediction of the likelihood of a PoI to be the next and its given likelihood of being next. This loss function is thus different from the previous one exploited by Ranking SVM where the loss is defined as the number of mis-ordered pairs of PoIs in a given ordering. More formally, let n be the size of $\hat{Y}$,

$$\Delta_{GBRT}\left(Y_f, \hat{Y}\right) = RMSE\left(Y_f, \hat{Y}\right) = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_f^i)^2 \tag{2}$$

where $\hat{Y}_i$ is the given likelihood of the i-th PoI in the correct ordering $\hat{Y}$ and Y_f^i is the prediction of the likelihood of the i-th PoI. The error computed on a single ordering as above is then averaged over all the orderings in the training set to obtain the total RMSE. GBRT is one of the current state-of-the-art approaches in *learning to rank*. In the Yahoo Labs Learning to Rank Challenge 2010 [Chapelle and Chang 2011] all winning methods used approaches that incorporated GBRT.

3.2. Features of PoIs and Tourist Trails

Our aim is to solve the LEARNEXT problem by learning from a given training set of trails the best function γ that can predict the ranking of PoIs, which a tourist has not yet visited, following their decreasing likelihood of being visited as the next PoI.

Learning to rank techniques employ a feature representation of the data to learn a prediction model. As a consequence, an important step for solving the LEARNEXT problem using *learning to rank* consists of carefully designing the feature space, so that the latent behaviour of the tourists contained in the data are captured and generalized during learning. This is a crucial phase that depends on the data used, from which we define the set of signals used for learning an effective prediction model.

In our tourism scenario we believe that many different dimensions influence how tourists choose PoIs in a city. When visiting a city, in fact, a tourist chooses what to visit next taking into account the popularity of a PoI, the distance of a given PoI with respect to her current position, how much a particular PoI matches her interests, the time needed to reach it, the time needed to visit it, etc. We identify a set of 68 different features, each one modeling one possible factor that a tourist can take into account for deciding what to visit. By employing these features, we are able to describe the trails of tourists as more than just PoI sequences thus giving them more significance than mere pattern occurrences. It is worth noting that tourist trails reflect how a city is made and how visitors see and perceive it during their visits. A tourist trail is, in fact, affected by distances between PoIs but at the same time by the type of tourist attractions, whether popular or specific.

We devise the signals above from a training set of trails that will be discussed later in Section 4. Each trail of the dataset is used to compute the 68 signals. To do it, we divide a trail of N PoIs in two parts: we assume the first $N-1$ PoIs to be the actual visit performed by the tourist (we refer to this part of the trail as “current path”) while the last PoI of the trail is used as candidate “next” PoI. Features can thus be broadly classified in two main categories, namely “Current Path” and “PoI” features. Roughly speaking, the two classes of features aim at modeling: i) the behavior of the tourist that can be distilled by what she has already seen in the actual current visit and ii) the characteristics of the candidate “next” PoIs.

Current path features are meant to model the tourist behavior during her actual/current visit (i.e., the sequence of PoI he has visited so far). They capture concepts on groups of visited PoIs like, for example, the total time that the tourist employed so far in her current path, the number of categories of the PoIs visited, the distances between them, some statistics about the length of the current path, etc.

The features belonging to this class aim at defining a “profile” of the tourist, by describing the many possible ways a tourist decides to move in a city. This is done by employing, for example, features representing aggregate information on distances between PoIs visited, i.e., is the distance very important when choosing a PoI to visit or is she driven also by other factors like the categories belonging to the PoI? How does a tourist visit a PoI? Does she do short or long visits? How much does she move in the city, i.e., does she show availability for moving or she tries to visit places very close each other? These questions lead us to the definition of this class of features. We add several features describing the behavior of the tourist from different perspectives: distance, transfer and visit time of PoIs, variety of categories belonging to PoIs visited, length of a path, total time spent in a path, etc.

PoI features model the characteristics of a candidate “next” PoI, also taking into account the past activities of other tourists. Accordingly, PoI features model the characteristics of the PoI to be suggested like, for example, the categories of the PoI, the popularity of the PoI, its distance w.r.t. the first PoI of the trail, etc.

The features belonging to this class aim at describing PoIs in terms of their characteristics. We believe the choice of choosing a particular PoI instead of another one (i.e., the “appealingness” of a particular PoI) is a function of its intrinsic characteristics and, more importantly, the characteristics it shows w.r.t. a particular trail we are considering. This observation lead us to define global characteristics as, for example, popularity, the categories that the PoI belongs, etc. Moreover, the second group of features are related to a particular trail we are considering, for example, the distance of the PoI w.r.t the first PoI of the current trail.

Tables I and II summarize the set of features we introduce and the intuition behind proposing such features, including their relationship to the current path or the next PoI or both. Current path features (Table I) are based on the current trail of a tourist. Given a trail, features can be, for example: the total transfer time and the actual visit time spent by a tourist, the number of unique categories of all PoIs in the trail, the euclidean and latitude/longitude distance of consecutively visited PoIs (average, max, min, total), the total time and length of the trail, the number of photos per PoI (average, max, min, total), and the length of the current paths belonging to the same tourist (average, max, min, total).

On the other hand, PoI features are based on the next PoI to be suggested and model, for example: the distance of the next PoI from the first PoI of the current path, whether the next PoI belongs to the top ten categories visited by tourists, the number of times a tourist visits that PoI in the training set, the conditional probability of observing that PoI given the last PoI visited by a tourist, the probability of observing the PoI as first (resp. last) PoI in the training set trails, the number of photos of the PoI (average, max, min), the number of past photos of the PoI from the same tourist, and the visit time of the PoI (average, max, min, total).

To sum up, the above presented features have been selected in order to describe the trails and PoIs in a city. However, they contribute to it in different ways. We perform a comprehensive analysis of this aspect in Section 4. Results shows that the influence of the features depends quite a lot on the type of city a tourist is visiting.

4. EXPERIMENTAL EVALUATION

In this section we present the main results of the paper. Firstly, we describe the dataset used for the evaluation of our solutions. We then present a comprehensive study aiming at answering the four new research questions we stated in Section 1.

4.1. Data Setup

To assess the effectiveness of our proposed techniques, we use three different datasets, built in a fully automatic process, by exploiting both photos from Flickr[1], a photo sharing portal, and Wikipedia pages. We build three datasets containing tourist movements covering three Italian cities important from a tourist point of view, namely Pisa, Florence and Rome. The rationale of the choice is to propose a complete evaluation of our techniques and its competitors by varying the size of the cities we are dealing with.

[1]http://www.flickr.com

Table I. List of current path features used to model the behavior of a tourist in a city.

Feature Name	Description	Intuition
actualTransferTime	Total transfer time from a PoI to the next one in a current path.	Time-based features are designed to understand how tourists exploit their available time to visit a city. For instance, the less the total transfer time the higher the chances that a tourist prefers closer PoIs.
actualVisitTime	The visit time for all PoIs in a current path, by summing up the time spent in each PoI.	
currPathTime	Total time for a current path, from beginning to end.	
categsPerCurrPath	Number of categories per current path.	A PoI might be associated with different categories. The more categories the more general a PoI is. This feature is designed to capture how much a tourist likes specific PoIs. For instance the diversity in tastes of a tourist can be used to trim down the possible candidate PoIs depending on their categories.
uniqueCategsPerCurrPath	The number of unique categories per current path.	
distLat_Avg distLat_Max distLat_Min distLat_Tot distLon_Avg distLon_Max distLon_Min distLon_Tot	Average, Max, Min, Total Latitude and Longitude distance between PoIs in a current path.	How spread out the already visited PoIs are. The less spread out the more likely the next PoI is going to be closer to the current one.
euclideanDist_Avg euclideanDist_Max euclideanDist_Min euclideanDist_Total	Average, Max, Min, Total Euclidean distance between PoIs in a current path.	
phPoICurrPath_Avg phPoICurrPath_Max phPoICurrPath_Min phPoICurrPath_Tot	Average, Max, Min, Total number of photos of the PoIs in a current path.	Information about number of PoIs (or information associated with them) and their importance (e.g. the number of photos per PoI) on the paths taken by a tourist in a given city. This is correlated (even if not exactly the same) with category information described above.
currPathLen	Number of PoIs in a current path.	
pathLen_Avg pathLen_Max pathLen_Min pathLen_Total	Average, Max, Min, Total length of all the paths belonging to a given tourist.	
currPathRatio	The ratio between the number of current paths performed by the current tourist and the maximum number of current paths per tourist, as observed in the training set.	This gives us an idea whether a tourist is active or not in comparison with the most active tourist.

The datasets have been made available for download to encourage the reproducibility of results[2].

We build the datasets by identifying the PoIs in a certain geographical region and the corresponding photos available on Flickr. Given an area of interest, we firstly collect

[2]Link to the trail datasets: http://hpc.isti.cnr.it/~nardini/datasets/LearNext.tar.gz

Table II. List of PoI features used to model the characteristics of each candidate destination.

Feature Name	Description	Intuition
`cat1, cat2, ..., cat10`	Top 10 most frequent categories.	Categories can be an indicator (combined with path features) of preferred PoIs for a user.
`numCategories`	The number of categories assigned to the next PoI.	
`distFromFirstPoI_Eucl` `distFromFirstPoI_Lat` `distFromFirstPoI_Lon` `distFromLastPoI_Eucl` `distFromLastPoI_Lat` `distFromLastPoI_Lon`	Latitude, Longitude and Euclidean distance from last and first PoI of the current path.	How far from the beginning of the trail has the tourist gone. The intuition is that the farther the tourist has gone the less likely she will move even farther.
`entropy`	The entropy of the last PoI of the current path.	This class of features entails aspects of the probability distributions associated with the considered next PoI. For instance the entropy captures the likelihood that from the current PoI one would almost always select among a very small subset of PoIs, therefore the smaller the entropy the smaller the set of candidate PoIs.
`middleProbab`	The probability that the next PoI is within a trail and not at the extremes.	
`startProb` `stopProb`	The probability that a PoI is first or last in a trail.	
`freqBigrams`	The frequency of the next PoI, given the last PoI of the current path.	
`freqTrigrams`	The frequency of the next PoI, given the last two PoIs of the current path.	
`noOfVisits`	The total number of visits of the next PoI in the collection.	Information on how likely a tourist will visit this PoI. Intuitively, the higher this value the more likely this PoI will be visited in general.
`ratioUsersVisitingPoI`	The ratio between the number of users visiting a given PoI and the total number of users.	
`numPhotos_Avg` `numPhotos_Max` `numPhotos_Min` `numPhotos_Total`	Average, Max, Min and Total number of photos of the next PoI in the collection.	Number of photos is another proxy for popularity. This group of features should represent, from a different perspective, how popular a PoI is among tourists.
`photosPerUser`	The total number of visits of the next PoI in the collection.	
`photosPoI_UserId_Avg` `photosPoI_UserId_Total`	Average and total no. of pics of the candidate PoI for a given user.	
`ratioPhotosPoI`	The ratio between the number of photos for the next PoI and the maximum number of photos per PoI, as observed in the training set.	
`ratioPoIInUserPhotos`	The ratio between the number of photos of the next PoI for a given tourist and the total number of photos taken by that user.	
`ratioTrailsWithPoI`	The ratio between the number of trails containing a given PoI and the total number of trails.	
`visitTimePoI_User`	The total visit time of a PoI for a given user.	Information about the visit time of PoIs. Intuitively, a tourist that has already spent lots of time in her trail will more likely visit PoIs requiring not too much time.
`visitTime_Avg` `visitTime_Max` `visitTime_Min` `visitTime_StdDev` `visitTime_Total`	Average, Max, Min, StdDev and Total visit time of a PoI.	

all the geo-referenced Wikipedia pages falling within this region. We assume each geo-referenced Wikipedia page, whose geographical coordinates falls into the given area, to be a PoI in the city we are analyzing. For each PoI, we retrieve its descriptive label as the named entity associated with it, its geographic coordinates as the ones specified in the Wikipedia page, and the set of categories the PoI belongs to, listed in the page[3].

[3]Wikipedia provides a fine-grained categorization of its geo-referenced pages.

The method is thus able to build a list of PoIs within a given geographical bounding box in a fully automatic way by exploiting Wikipedia as an external source of knowledge.

To devise tourist trails in the area of interest we query Flickr to retrieve the metadata (user id, timestamp, tags, geographic coordinates, etc.) of the photos taken in the given area. The assumption we are making is that photo albums made by Flickr users implicitly represent tourism itineraries within a given city.

To strengthen the assumption and thus the accuracy of our method, we retrieve only photos having the highest geo-referenced precision in the given area of interest. Then, we collect geo-tagged photo albums (i.e., sets of photos logically grouped under a common name) from Flickr users. We discard photo albums containing only one photo and those containing photos with no GPS information associated. Eventually, photos are mapped to the set of PoIs previously collected from Wikipedia. This is done by associating a photo to a PoI if that photo is in the circular area having the PoI as its center and $r = 100$ meters as its radius. Moreover, since several photos by the same user are usually taken close to the same PoI, we collapse them by considering the timestamps associated with the first and last of these photos as the starting and ending time of a user visit in a PoI. The results of the assignment above produce, for each Flickr user, a stream of PoIs she visited. To sum up, given a geo-tagged photo album with at least two photos, we are able to build a trail by mapping photos to PoIs and collapsing them into trails of PoIs visited by Flickr users.

Finally, in order to build the trail sets, we need a way to split the stream of PoIs visited by each tourist in a meaningful and realistic time-wise set of trails. We employ a time-based cutting method that produces the list of trails a tourist performed, by considering the inter-arrival time of each pair of sequential photos in her stream. To do so, for each city, we compute the probability distribution of the inter-arrival time x to be less then a given time threshold k, i.e., $P(x \le k)$. Then for each dataset we devise the time threshold k corresponding to $P(x \le k) = 0.9$. Regarding Rome, it corresponds to 5 hours, for Florence 6 hours, while for Pisa 3 hours. A similar methodology is employed also in [Brilhante et al. 2013], as both works are based on the same dataset, although presenting minor specific changes.

Table III shows the main properties of the datasets we use to evaluate our techniques. We report the number of PoIs (column "# PoIs") that have been found for each of the three cities. Furthermore, columns "# Users", "# Photos" and "# Trails" report the number of distinct users, the number of public photos we crawl from Flickr, and the total number of trails extracted from the photos in the dataset. Table IV shows the main properties of the trails. We report the number of trails (i.e., longer than two PoIs), the number of PoIs visited at least once and the average number of trails going through each PoI.

Table III. Properties of the three datasets.

Dataset	# PoIs	# Users	# Photos	# Trails
Pisa	124	1,825	18,170	3,446
Florence	1,022	7,049	102,888	16,863
Rome	671	13,772	234,616	35,602

Table IV. Properties of the tourist trails in the three datasets.

Dataset	# Trails ≥ 2	# Visited PoIs	Avg. Trails per PoI
Pisa	992	110	9.01
Florence	5,984	888	6.73
Rome	12,565	490	25.64

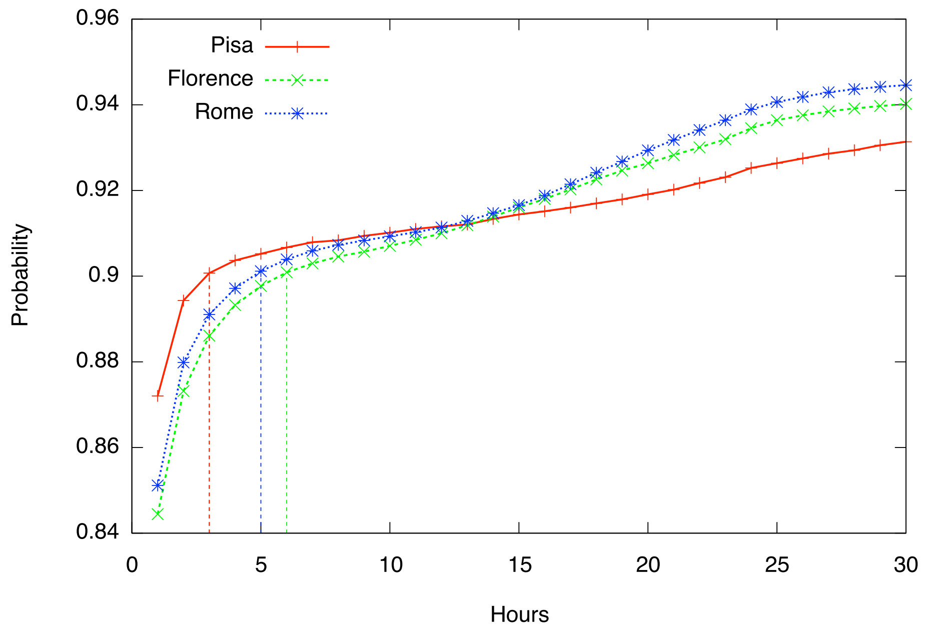

Fig. 1. Probability distribution of the inter-arrival time for each set of photos of the three cities analyzed. The horizontal dotted lines highlight the time thresholds used to cut the trails. The leftmost one (3 hours) has been used to cut the trails for Pisa, the central one (5 hours) refers to the threshold used to produce trails for Rome, and the rightmost one (6 hours) has been used to cut trails for Florence.

Given a PoI, the set of possible destinations is an important information we may exploit in order to detect the most likely next PoI a tourist will visit. Figures 2(a), 2(b), 2(c), 2(d) present important properties of the dataset. Figure 2(a) and 2(b) reports the distribution of trails and PoIs in the dataset, while in Figure 2(c) we plot the outlink distribution of PoIs for the three cities. All of them are power-law. Figure 2(d) shows the outlinks entropy of the three datasets computed on the distribution of PoIs reached from the previous ones. In this case the lower the entropy the higher the likelihood that a tourist will select a frequently visited PoI.

4.2. Effectiveness Evaluation

The evaluation of our solution to the LEARNEXT problem is aimed at answering the following research questions[4]:

— Are *learning to rank* techniques effective for predicting the next PoI? (RQ1)
— What is the robustness of the proposed techniques? In particular, how does the performance change by varying the learning parameters? (**RQ2**)
— How does the length of trails affect the effectiveness of the proposed models? (**RQ3**)
— How does the popularity of PoIs affect the prediction? (**RQ4**)
— What is the sensitivity of the model with respect to the proposed features? How does the performance of the model change by varying the set of features employed? (**RQ5**)

We intend to answer the questions above by adopting a standard training/test evaluation strategy over the three datasets of available trails. For each of the three cities,

[4]We highlight in bold the new research questions that have not been previously addressed in [Baraglia et al. 2013].

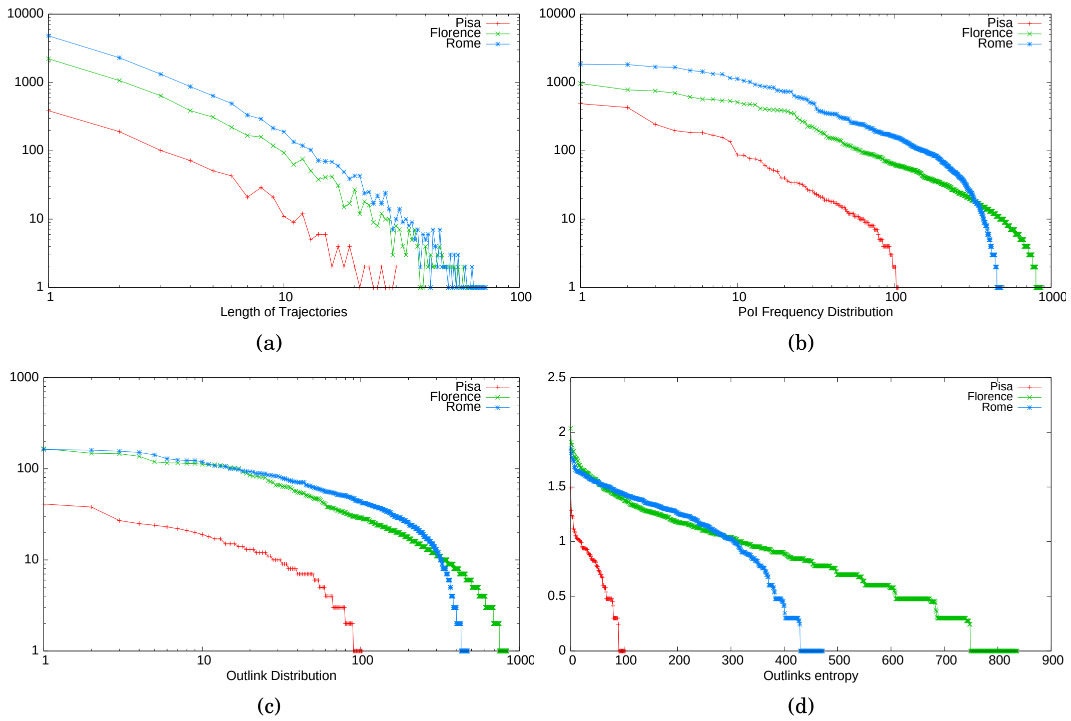

Fig. 2. Properties of the datasets used: (a) shows the distribution of the trail length, (b) shows the distribution of the PoI frequency, (c) shows the outlink distribution of the PoIs and (d) shows the distribution of the outlinks entropy of the PoIs.

we generate a training set (80%) and a test set (20%) of trails. The effectiveness of the methods is assessed by means of Success@k (i.e., the percentage of times that the correct answer is in the top-k ranked PoIs [Pennacchiotti et al. 2012]), Mean Reciprocal Rank (MRR@k), and total Mean Reciprocal Rank (MRR) [Baeza-Yates et al. 1999]. MRR is defined as the reciprocal rank of a prediction, i.e., the multiplicative inverse of the rank of the correct answer. Moreover, we compare our solutions against a probability baseline and four important state-of-the-art techniques, i.e., WhereNext [Monreale et al. 2009] and Random Walk [Lucchese et al. 2012], logistic regression and SVM classifier. All the methods have been tested by using the same datasets and the same training/test methodology described above. The results have been validated by means of a standard 10-fold cross validation.

For comparing our solution to the state of the art we use five different baselines. More in detail, "PROB" is a pure baseline modeling the markovian behavior of the tourist, "WhereNext" and "Random Walk" are two well-known state-of-the-art methods for predicting tourist movements, while "Logistic Regression" and "SVM-C" are two well-known state-of-the-art techniques for regression/classification. Different reasons led us to compare our proposals against these five methods. We report them below together with a description of the methods.

— "PROB" uses the training set to build a directed graph where nodes are PoIs of a given city and edges are transactions from a source PoI to a destination PoI. Each edge is weighted with the probability to observe the transaction from a source PoI to a destination PoI (if any) in the training set. Given a PoI currently visited by a tourist, PROB predicts the most likely PoI to be visited next by selecting from the set of the current PoI's outlinks, the one with highest probability. The reason behind comparing LEARNEXT against "PROB" is to understand what the difference is between our method exploiting 68 different features and a simple method exploiting only one (even if very strong) signal. This comparison thus allows us to understand the "real" gain against methods employing simple statistics to predict the next PoI.

— "WhereNext" [Monreale et al. 2009] uses trajectory pattern mining to devise T-Patterns, i.e., frequent behaviors of movement in a city from data. T-Patterns constitute the knowledge model used to compute the prediction. In this paper, we use the original implementation of the predictor presented in [Monreale et al. 2009] kindly provided to us by the authors. We test different combinations of parameters to mine T-Patterns from our datasets. We compare LEARNEXT against "WhereNext" because the latter one is based Trajectory Pattern Mining. The method is known to be effective in predicting locations. Moreover, it can be applied to the dataset we build from Flickr. For these reasons we believe it is a good competitor for LEARNEXT.

— "Random Walk" [Lucchese et al. 2012] employs a graph-based representation of the PoIs in a city. Authors named it "itinerary graph" and exploit it by using a random walk with restart to select the most relevant PoIs for a given tourist. As for WhereNext, we use the original implementation of the method presented in [Lucchese et al. 2012], provided by the authors. We build the itinerary graph over each of the cities we are considering and, for each trail in the test set we compute the list of the top-10 recommendations. The list of recommended PoIs is then used to evaluate the effectiveness of the method for predicting the next candidate PoI. For the two methods above, we report only the best performance obtained. We compare LEARNEXT against "Random Walk" because it is tailored for tourism. It employs Flickr and Wikipedia data as LEARNEXT. It is also known to be an effective system for prediction/recommendation of tourist locations.

— "Logistic Regression" [Bishop et al. 2006] is used to predict the probability of a PoI being relevant (i.e., the next one) or not relevant for a certain current path. We use this method because logistic regression is an effective technique for classification capable of approximating the predicted classes with a certain probability. Instead of outputting a class for a given feature vector, it gives the probability that the vector is relevant or not, thus we can compare it to the ranking offered by the ranking models.

— "SVM-C" [Cortes and Vapnik 1995] uses Support Vector Machines for classification and is used similarly to "Logistic Regression". SVM is used to binary classify PoIs in two classes: "relevant", "not relevant", but also set to output probabilities, thus comparable to our proposed methods.

The evaluation strategy we use to assess how the proposed techniques behave in terms of effectiveness (RQ1) is the following: each model for the three cities has been trained on the corresponding training set. A training set contains positive and negative examples of a candidate next PoI, represented by its features. Given a trail of length N, the training set contains both current path features (computed on the first $N - 1$ PoIs of the trail) and PoI features (computed for the N-th PoI). The latter are computed considering both the actual next PoI visited by the tourist, i.e., the N-th PoI of the trail (as a positive example) and a few negative examples, with PoIs different from the ones seen in the actual trail.

We experiment the performance of LEARNEXT by varying the negative examples in the training set. We conduct this analysis in the city of Florence and we experiment 1, 3, 5, and 10 negatives examples per trail. Negative examples have been sampled on a distance basis. The rationale behind this choice is to allow LEARNEXT to be robust toward possible false positives that are close to the current PoI visited by the tourist. We do so by selecting as negative examples PoIs close to the N-th one in the trail while one has been selected far from the N-th one.

Results, in term of Success@1 show the following performance in the case of Florence: 34.92% with 1 negative example, 37.76% with 3 negative examples, 36.84% with 5 negative examples and 37.51% with 10 negative examples. In the case of 1 negative example the performance is slightly below than in the case of 3 or more negatives, meaning the algorithm produces better results when provided with more learning examples, while at the same time not creating a negative bias. Moreover, it is worth highlighting that there is no statistical significance between the results obtained by using 3 or more negative examples. This analysis led us to use 3 negatives examples, so as not to create a negative bias in the resulting learned models but still feed the algorithm with enough data for computing the ranking information. This choice also allow us to have a lighter process for building the training set, than when using more negative examples.

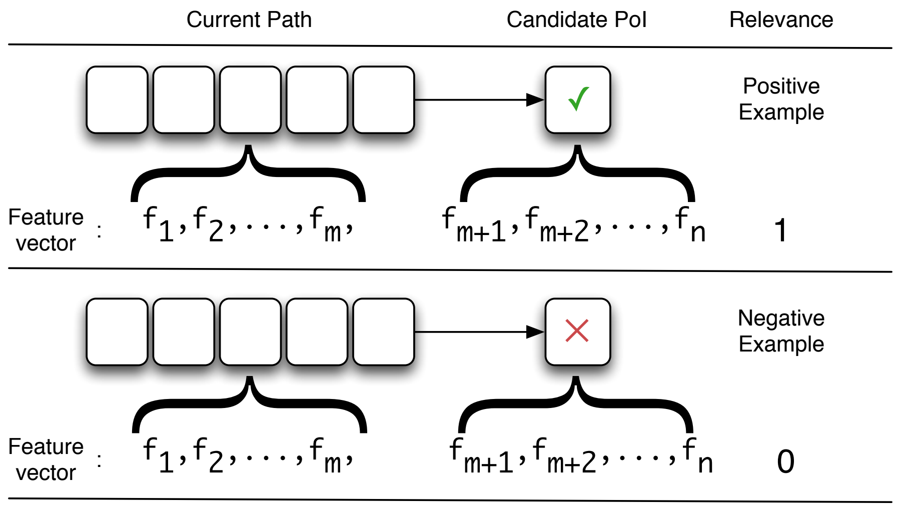

Fig. 3. Assignment of relevance labels to sequences of PoIs and extraction of the feature vectors for corresponding trails. If a PoI is actually visited in the training set, its relevance label is set to 1, otherwise it is set to 0.

For building the test set we adopt the following process. Given a trail of length N in the test set, we use the first $N - 1$ PoIs of the trail to profile the tourist history and re-rank all final PoIs observed in the training, according to the prediction model. The resulting sorted list is then evaluated by using the metrics introduced previously. The aim of this evaluation is thus to measure how many times our models are able to re-rank correctly, i.e., to rank in the first positions of the whole list of PoIs the actual next PoI.

Answering Research Question #1

RQ1: Are *learning to rank* techniques effective for predicting the next PoI?

In the first experiment we measure metrics like: Success@k, MRR@k and total MRR, i.e., MRR computed on the complete list of re-ranked PoIs. Results are computed for all the techniques, our proposed solutions to the LEARNNEXT problem along with three methods we choose as baselines. Table V shows the results of the experiment.

WhereNext and Random Walk never outperform PROB in terms of Success@1. The techniques we propose instead outperform all the competitors. Regarding Pisa, in terms of Success@1, Ranking SVM scores 37.18% and GBRT scores 40.70%, Logistic Regression scores 29.14%, SVM-C scores 26.13% and PROB scores 16.08%. As expected, Logistic Regression is the most effective competitor while lower performances can be highlighted for SVM-C. However, our techniques always outperform the baseline by at least 10% in the case of Pisa, 7% in the case of Florence and 6% in the case of Rome. An important note is that we offer results for PROB only @1 for reasons of fairness, as it extracts candidates only from outgoing links, whereas the other methods reorder all PoIs in each dataset.

Important results should be highlighted also for Success@2. Here, our methods are able to score 52.76% (Ranking SVM), and 55.27% (GBRT). In half of the cases our methods are able to rank the actual next PoI in the two highest positions of the list. Performance improves when considering higher values for the cut-off parameter. In particular, if we look at the performance in terms of Success@10, Random Walk attains a score of 46.73%, Logistic Regression scores 55.77% and SVM-C scores 50.25% whereas Ranking SVM scores 76.38%, and GBRT scores 88.44%. GBRT is the technique showing the best performance, while Ranking SVM is second and both techniques perform better than the competitors we considered. Furthermore, the result for total MRR points out that, even if Ranking SVM and GBRT in some cases are not able to place the next PoI in a high position of the list, the overall ranking does not degrade significantly. The same behavior could be highlighted for the medium and the large city, where both Ranking SVM and GBRT are always outperforming the competitors. In particular, while for Firenze GBRT performs about 7 times better w.r.t. PROB, when considering Rome GBRT only doubles the performance of PROBand the same trend is confirmed for both Logistic Regression and SVM-C. Looking at the Entropy plotted in Figure 2(d) we would expect that PROB would perform better for Pisa than for Florence and Rome. Indeed, from results in Figure 2(d) we can observe that PROB behaves as expected. Nevertheless, entropy distribution is very skewed and we expect that in many cases probability features are not enough for a good performance. This is confirmed once again by the effectiveness of our learning based techniques.

From the results shown above we conclude that RQ1 has an affirmative answer: *learning to rank techniques are effective for predicting the next PoI in a trail*.

Table V. Effectiveness (%) in terms of Success@k, MRR@k, and total MRR of the proposed techniques along with the competitors.

City	Predictor	Success (MRR)					MRR
		@1	@2	@3	@5	@10	
Pisa	Prob	16.08%	-	-	-	-	-
	WhereNext	12.56%	-	-	-	-	-
	Random Walk	15.07% (0.15)	20.60% (0.17)	25.12% (0.19)	31.65% (0.20)	46.73% (0.22)	-
	Ranking SVM	37.18% (0.37)	52.76% (0.44)	60.30% (0.47)	69.34% (0.49)	76.38% (0.50)	0.51
	GBRT	40.70% (0.40)	55.27% (0.47)	63.81% (0.50)	75.87% (0.53)	88.44% (0.55)	0.56
	Logistic Regression	29.14% (0.29)	36.18% (0.32)	39.19% (0.33)	45.22% (0.35)	55.77% (0.36)	0.37
	SVM-C	26.13% (0.26)	28.64% (0.27)	33.66% (0.29)	39.69% (0.30)	50.25% (0.31)	0.33
Florence	Prob	4.59%	-	-	-	-	-
	WhereNext	2.90%	-	-	-	-	-
	Random Walk	3.25% (0.03)	6.09% (0.04)	8.77% (0.05)	11.69% (0.06)	20.13% (0.07)	-
	Ranking SVM	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42
	GBRT	37.76% (0.37)	46.78% (0.42)	53.04% (0.44)	59.31% (0.45)	69.34% (0.47)	0.48
	Logistic Regression	29.40% (0.29)	35.00% (0.32)	36.59% (0.32)	39.18% (0.33)	42.52% (0.33)	0.34
	SVM-C	27.73% (0.27)	31.41% (0.29)	32.16% (0.29)	34.50% (0.30)	36.34% (0.30)	0.31
Rome	Prob	12.93%	-	-	-	-	-
	WhereNext	6.37%	-	-	-	-	-
	Random Walk	8.43% (0.08)	13.76% (0.11)	19.22% (0.12)	26.38% (0.14)	38.12% (0.16)	-
	Ranking SVM	25.06% (0.25)	32.51% (0.28)	37.24% (0.30)	48.22% (0.32)	61.99% (0.34)	0.35
	GBRT	30.95% (0.30)	40.07% (0.34)	47.15% (0.38)	56.34% (0.40)	67.68% (0.41)	0.42
	Logistic Regression	24.07% (0.24)	30.24% (0.27)	33.82% (0.28)	38.95% (0.29)	46.27% (0.30)	0.31
	SVM-C	19.77% (0.19)	23.27% (0.21)	24.43% (0.21)	26.82% (0.22)	30.32% (0.22)	0.23

Answering Research Question #2

RQ2: What is the robustness of the proposed techniques? In particular, how does the performance change by varying the learning parameters?

We evaluate the prediction performance of both Ranking SVM and GBRT by varying the parameters used in the learning phase. We have done grid search on a portion of the training set and validated it on a validation set (held out again from the training). When building the GBRT model, we vary both the "number of leaves" (NL) used in the model and the "learning rate" (LR), while for Ranking SVM we vary the cost parameter (C) as it controls soft margins needed for avoiding misclassifications. In particular, NL assumes the values in $\{7, 10, 15, 20\}$, LR assumes values in $\{0.01, 0.05, 0.1\}$, while C assumes values in $\{3, 6, 10, 30, 50, 100, 200, \ldots, 1,000\}$. Tables VI and Figure 4 report the different performance of our proposed solutions.

The best performance of the GBRT technique for Pisa is reached on two different sets of parameters: (NL = 20, LR = 0.01) and (NL = 15, LR = 0.05). However, we highlight the best one in respect to the overall performance (i.e, by considering also Success@k, $k > 1$). Moreover, the best performance of GBRT for Firenze is achieved by using 15 leaves and learning rate equal to 0.05. A smaller number of leaves (10) and a smaller learning rate (0.01) allows to obtain the best performance when dealing with data from Rome.

We also study the statistical significance of the results we obtain in this analysis. In particular, we test whether the difference between results is statistically significant w.r.t. different parameter combinations. We analyze the difference between the maximum value and various results for each city obtained by varying the number of leaves and the learning rate. If the difference is statistically significant we highlight it with a ▼, whereas if the difference is not statistically significant we use a ▽. The test is conducted by setting p-value $= 0.05$. As shown in Table VI there is no statistical significance between the maximum performance and other various parameter settings in the case of Pisa and Florence, while for Rome results are different. A possible explanation of this phenomenon relies in the different sizes of the populations between the three cities covered by the dataset.

Table VI. Effectiveness (in terms of Success@1) of the models by varying the learning parameters of the GBRT. In bold we highlight the best performance.

City	NL	LR		
		0.01	0.05	0.1
Pisa	7	38.69%	38.69% ▽	36.18%
	10	38.19%	38.19% ▽	37.18%
	15	39.69% ▽	**40.70%**	33.16% ▽
	20	40.70%	37.68% ▽	33.66%
Florence	7	35.33%	35.58% ▽	33.24%
	10	36.75%	37.09% ▽	35.75%
	15	37.00% ▽	**37.76%**	36.00% ▽
	20	36.50%	37.25% ▽	36.84%
Rome	7	30.48% ▽	30.36%	28.25%
	10	**30.95%**	29.88% ▽	29.36% ▽
	15	27.01% ▼	29.40%	28.01%
	20	27.61% ▼	29.00%	28.65%

We also study how Ranking SVM varies its performance w.r.t. the parameter C. Concerning Pisa, the best behavior @1 is achieved for $C = 600$, $C = 800$ and $C = 1000$, whereas $C = 1000$ shows the best overall performance by varying k. Moreover, Figure 4 shows that, for Florence and Rome the optimal performance is reached when $C = 200$ and $C = 500$, respectively. From the results we can devise that the differences of the size and particularities of the datasets needs a differentiation of the parameter C. However, as we increase the value of C, (i.e., $C > 500$) results tend to stabilize with no statistically significant difference between the achieved performance.

Consequently, the answer to **RQ2** is that, *for the proposed techniques, a careful tuning of the parameters must be done on the basis of the size of the dataset considered*.

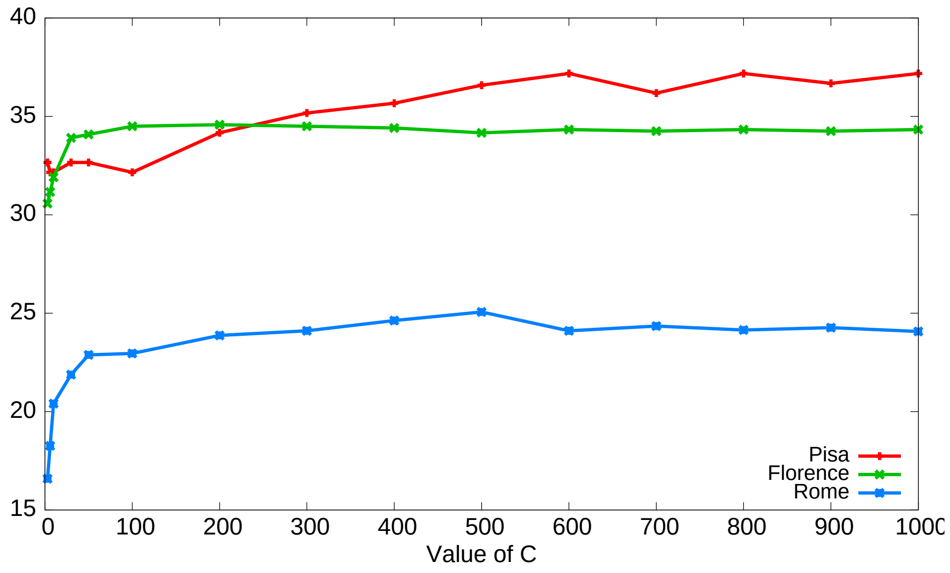

Fig. 4. Efficiency of Ranking SVM by varying parameter C.

Answering Research Question #3

RQ3: How does the length of trails affect the effectiveness of the proposed models?

We assess how the length of trails affects the performance of our technique. This is an important aspect to study as our feature space is made of two subsets of features: current path and PoI features. While PoI features model the characteristics of a PoI, current path features model the behavior of the tourist when visiting the city. This behavior is derived from the PoIs she already visited in the trail. Thus, current path features strictly depend on the set of PoIs already visited (we refer to it as “history”). In this section we want to understand what is the influence of shortening training trails on the prediction quality of the learned model. In order to do that we build a training set of trails by varing the number of PoIs exploited to derive current path features. To answer to **RQ3**, we thus evaluate the prediction of our models by varying the number of PoIs used to compute the set of current path features. To do so, for each dataset available, we define the maximum number of PoIs constituting the current path of a given tourist. As an example, $k = 3$ means that the first three PoIs are used to compute the current path features in the training data.

Adding more information to the history (in terms of PoIs visited) determines an important gain of the predictive performance of the proposed methods. This gain is clear when comparing the performance obtained by exploiting 1 PoI and 3 PoIs for computing current path features. If history contains only 1 PoI, the method loses its performance w.r.t. PROB. A possible explanation of this phenomenon relies in the complexity of the models behind LEARNEXT. Such models are complex formulae obtained by combining 68 different signals. Their normal behavior is thus to drop performance if a set of such signals lose their expressiveness regarding previous tourist interactions. To conclude, the behavior highlighted by the reviewer is expected. If a set of features exploited by the model are missing, the model is not able to rank PoI thus resulting in a very low prediction power. The impact of the history on prediction performance can be seen in Figure 5, where we test the effectiveness of our solutions by varying the number of PoIs used to derive current path features.

The answer to **RQ3** is that *historical information has an important role in the definition of a model for predicting a PoI a tourist will visit.* Effectiveness measured in terms of Success@1 shows a steady growth for a history size depending on the city size. In the case of small cities (e.g., Pisa), having a longer history helps in better understanding the particular tastes of tourists. In the case of medium and big cities (e.g. Florence and Rome), history length is important to a lesser extent. This may be due to the fact that, for small cities, the entropy of tourists' decisions is lower. Therefore, we need more contextual information in order to improve predictions in respect to those made by taking into account only probability features.

A related experiment to this one aims at understanding what is the effectiveness of the model by varying the number of PoIs used to derive current path features from test trails. In this kind of test, we maintain the full predictive power of the initial training (the one obtained by exploiting the full history of the trails) and we vary the history of the test trails. This experiment reflects the predictive power of the model when a tourist is at the beginning of her visit and how it changes by adding more visited PoIs to her visit. As expected, results improve as the length of test trails increases.

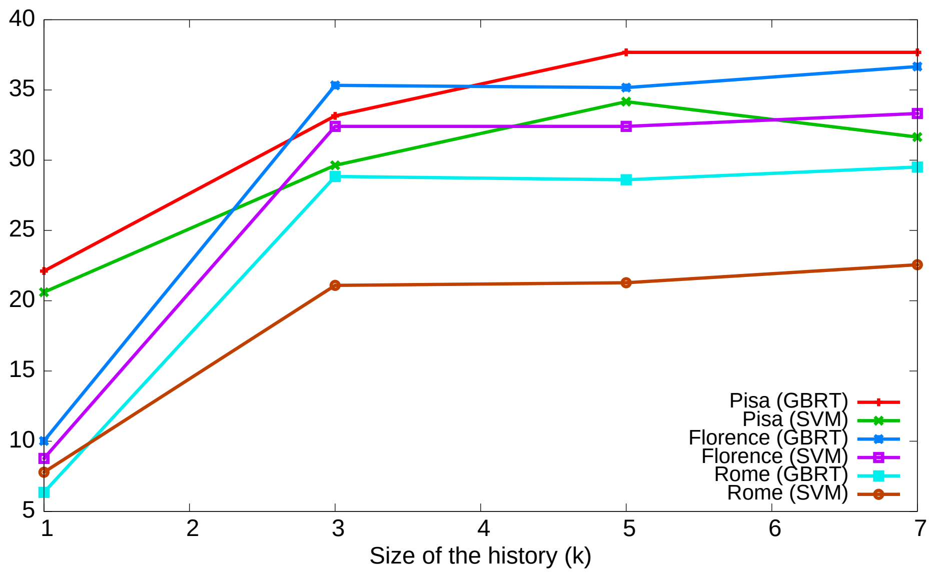

Fig. 5. Effectiveness (in terms of Success@1) of the proposed methods by varying the size of the history used to train/test the model.

Answering Research Question #4

RQ4: How does the popularity of PoIs affect the prediction?

We aim at understanding how our proposed techniques behave by taking into account the popularity of the PoIs to be suggested. In particular, we are interested in understanding how the visit frequency of PoIs impacts on the models behavior. We assess it by sampling, from each of the three test sets, two distinct sets of trails, those whose N-th PoI (to be predicted) is observed to be either frequent or rarely visited in the training set. We consider the top-10 most frequent PoIs for each dataset. The trails with the N-th PoI included in top ten belong to the frequent last PoI test set, while the remaining ones belong to the rare last PoI test set. To be as fair as possible we evaluate our PROB baseline in the same way.

As it can be seen in Table VII, there is a substantial difference amongst frequent and rare PoI prediction results. For Pisa, PROB captures the frequent PoIs, with a 15.07% in Success@1, while offering a poor performance in the case of rare PoIs, 1% Success@1. This is due to the specific properties and size of the dataset, which succeeds in capturing frequent passages. However, these characteristics change when dealing with larger datasets corresponding also to larger cities. In the case of Florence and Rome, due to their topology, the performance is quite similar, but still low, namely 2.50% and 6.20% Success@1 for frequent PoIs and 2.17% and 6.80% Success@1 for rare PoIs.

On the other hand, the proposed methods - Ranking SVM and GBRT - show an improvement for both frequent and rare PoIs. For Pisa, the GBRT Success@1 for frequent PoIs rises up to 47.27%, while Success@10 rises up to 95.45%. We notice also a steady growth in the MRR, with an overall 0.63%. The values in the case of Ranking SVM are slightly lower than GBRT, but still 3 times larger than PROB. A significant improvement with respect to PROB can be seen also in predicting rare PoIs. For Florence, results show an even bigger improvement w.r.t. that described for Pisa. In particular, the prediction of frequent PoIs in this case reaches 50.84% for GBRT and 44.53% for Ranking SVM, respectively.

The answer to **RQ4** is that *prediction models are better at predicting popular PoIs rather than rarely visited PoIs. Our proposed solutions based on learning to rank models reach high prediction performance also in the case of less popular PoIs*.

Table VII. Effectiveness of the proposed techniques on predicting frequently (F) and rarely (R) visited PoIs. We report both Success@k, MRR@k and total MRR.

City	Predictor	Success (MRR)					MRR
		@1	@2	@3	@5	@10	
Pisa (F)	PROB	15.07% (0.15)	-	-	-	-	-
Pisa (F)	GBRT	47.27% (0.47)	62.72% (0.55)	73.63% (0.58)	85.45% (0.61)	95.45% (0.62)	0.63
Pisa (F)	Ranking SVM	45.45% (0.45)	64.54% (0.55)	75.45% (0.58)	82.72% (0.60)	88.18% (0.61)	0.61
Pisa (R)	PROB	1% (0.01)	-	-	-	-	-
Pisa (R)	GBRT	32.58% (0.32)	46.06% (0.39)	61.58% (0.41)	64.04% (0.44)	79.77% (0.46)	0.47
Pisa (R)	Ranking SVM	16.85% (0.16)	31.46% (0.24)	31.46% (0.24)	44.94% (0.27)	55.05% (0.28)	0.30
Florence (F)	PROB	2.50% (0.02)	-	-	-	-	-
Florence (F)	GBRT	50.84% (0.50)	64.28% (0.57)	71.84% (0.60)	79.41% (0.61)	84.87% (0.62)	0.63
Florence (F)	Ranking SVM	44.53% (0.44)	53.78% (0.49)	61.34% (0.51)	69.32% (0.53)	78.57% (0.54)	0.55
Florence (R)	PROB	2.17% (0.02)	-	-	-	-	-
Florence (R)	GBRT	34.51% (0.34)	42.44% (0.38)	48.38% (0.40)	54.32% (0.41)	65.58% (0.43)	0.44
Florence (R)	Ranking SVM	27.84% (0.27)	33.47% (0.30)	36.07% (0.31)	38.99% (0.32)	44.00% (0.32)	0.34
Rome (F)	PROB	6.20% (0.06)	-	-	-	-	-
Rome (F)	GBRT	45.95% (0.45)	63.00% (0.54)	71.67% (0.57)	82.65% (0.59)	90.89% (0.61)	0.61
Rome (F)	Ranking SVM	36.27% (0.36)	51.87% (0.44)	60.83% (0.47)	77.60% (0.50)	93.06% (0.52)	0.53
Rome (R)	PROB	6.80% (0.06)	-	-	-	-	-
Rome (R)	GBRT	23.06% (0.23)	30.14% (0.26)	36.79% (0.28)	45.24% (0.30)	58.20% (0.32)	0.34
Rome (R)	Ranking SVM	16.41% (0.16)	22.02% (0.19)	27.07% (0.20)	35.31% (0.22)	46.73% (0.24)	0.25

Answering Research Question #5

RQ5: What is the sensitivity of the model with respect to the proposed features? How does the performance of the model change by varying the set of features employed?

We further analyze the features with which we describe our data and their predictive power. We do this in three different ways. First, we investigate the relationships between each feature and the associated target labels in the training set. We propose it as a preliminary analysis because the method studies how much single features are related with the associated label. The method does not consider possible relations between features. To extend the results of the first analysis, we resort to analysing the structure of the models (i.e., forests of trees) generated by GBRT. As further described, this analysis is more effective in understanding the importance of a feature because it takes into account possible dependencies between features. Finally, we study the pairwise mutual relationships between features to derive the relations between them and thus to find the most relevant ones by choosing good representatives from groups of similar features. We do it by employing a graph-based model of the relationships between them. Lastly, we compare our findings against several well-known techniques for feature selection.

Relationships between features and labels. We compute the Pearson product-moment correlation coefficient (r) between each feature and the class of each PoI in the training set. Pearson's r indicates *a priori* the features that weight most in the model. We are interested in medium/high correlations ($-1 \leq r \leq -0.3$ and $0.3 \leq r \leq 1$) between a given feature and the corresponding class/target of a given PoI. The results are presented in Table VIII.

We obtain three different set of features with medium/high correlation w.r.t. the label of PoIs in the training set. Each feature in the three sets captures a concept of "popularity" of the PoI determined by means of different approaches. As an example, while `noOfVisits` measures the number of times a PoI is visited (the higher the number, the higher the popularity thus showing a positive correlation), `ratioPoIInUserPhotos` measures how popular a PoI is in each user's photo collection. Moreover, `numPhotos` reflects the popularity of a PoI in the collection and `ratioUsersVisitingPoI` measures the ratio of users in the training set that visit a PoI. In the case of Pisa, an interesting anti-correlation is shown by `distFromLastPoI_Eucl`. This feature measures how far is the candidate PoI w.r.t. the last PoI that has been visited by a tourist: the closer the PoI, the more likely for it to be visited next. This suggests that, in small cities, the behavior of a tourist is mainly affected by distance-based criteria. `freqBigrams` and `freqTrigrams` capture a more structured concept of popularity as they measure how many times a PoI is visited given the previous one or two PoIs, respectively. Results from this experiment highlights that the popularity of a PoI is an important signal to take into account by tourists when choosing the next PoI to visit.

Importance of the features assessed by GBRT models. A different analysis we conduct is that of studying the importance of the features in a given GBRT-based model by computing the average position of each feature within the model, namely the average position of a feature within the regression trees ensemble. This analysis piggybacks the behavior of GBRT-based learning techniques in selecting the features to be used as splitting nodes [Pan et al. 2009]. These techniques build the model by selecting at each iteration the feature that minimizes a given error metric. GBRT are a particular class of regression trees that exploit gradient boosting for minimizing RMSE. The average position of a feature in the model thus reveals its importance in approximating the training dataset. The more the average position of a feature occurs higher in the model, the more it is discriminative in describing the training set.

Table VIII. Features per dataset showing high or moderate Pearson's correlation coefficient w.r.t. the training label. The analysis is performed per city on the models showing the best prediction performance.

Pisa		Florence		Rome	
Feature	Correlation	Feature	Correlation	Feature	Correlation
`ratioUsersVisitingPoI`	0.492	`freqTrigrams`	0.501	`ratioPhotosPoI`	0.440
`ratioTrailsWithPoI`	0.489	`visitTime_Total`	0.342	`numPhotos_Total`	0.440
`freqBigrams`	0.484	`ratioPhotosPoI`	0.323	`freqTrigrams`	0.425
`noOfVisits`	0.481	`numPhotos_Total`	0.323	`freqBigrams`	0.403
`ratioPhotosPoI`	0.466	`visitTime_Max`	0.315	`noOfVisits`	0.368
`numPhotos_Total`	0.466			`ratioTrailsWithPoI`	0.363
`numPhotos_Max`	0.440			`ratioUsersVisitingPoI`	0.362
`visitTime_Total`	0.431			`numPhotos_Max`	0.315
`visitTime_Max`	0.423				
`freqTrigrams`	0.419				
`ratioPoIInUserPhotos`	0.345				
`distFromLastPoI_Eucl`	-0.374				

Table IX. Top-10 features per dataset with a high average position in the GBRT model. The analysis is performed per city on the models showing the best prediction performance (see Table VI).

Pisa		Florence		Rome	
Feature	AvgPos. (Occ.)	Feature	AvgPos. (Occ.)	Feature	AvgPos. (Occ.)
freqBigrams	2.914 (129)	freqBigrams	2.700 (172)	freqBigrams	2.385 (614)
freqTrigrams	5.085 (83)	freqTrigrams	6.927 (88)	freqTrigrams	3.155 (331)
distFromLastPoI_Eucl	9.928 (60)	entropy	9.234 (122)	numPhotos_Total	6.997 (213)
distFromLastPoI_Lat	10.528 (45)	visitTime_Max	10.117 (63)	ratioPhotosPoi	7.197 (206)
ratioPoIInUserPhotos	10.742 (33)	ratioPoIInUserPhotos	10.837 (41)	numPhotos_Avg	7.586 (230)
distFromLastPoI_Lon	10.871 (36)	distFromFirstPoI_Eucl	10.927 (54)	ratioPoIInUserPhotos	7.661 (122)
visitTime_Avg	11.057 (38)	noOfVisits	11.234 (71)	currPathLen	7.954 (182)
startProb	11.157 (38)	startProb	11.432 (58)	visitTime_Max	8.185 (182)
visitTime_Max	11.242 (35)	distFromLastPoI_Eucl	11.468 (62)	noOfVisits	8.248 (167)
entropy	11.257 (37)	uniqueCategsPerCurrPath	11.531 (52)	stopProb	8.448 (107)

Table X. Top-10 features per dataset with the highest number of occurrences in the GBRT model. The analysis is performed per city on the models showing the best prediction performance (see Table VI).

Pisa		Florence		Rome	
Feature	Occurrences	Feature	Occurrences	Feature	Occurrences
freqBigrams	129	freqBigrams	172	freqBigrams	614
freqTrigrams	83	entropy	122	freqTrigrams	331
distFromLastPoI_Eucl	60	freqTrigrams	88	numPhotos_Avg	230
distFromLastPoI_Lat	45	noOfVisits	71	numPhotos_Total	213
visitTime_StdDev	40	numPhotos_Avg	64	ratioPhotosPoi	206
startProb	38	visitTime_Max	63	currPathLen	182
visitTime_Avg	38	distFromLastPoI_Eucl	62	visitTime_Max	182
entropy	37	startProb	58	noOfVisits	167
distFromFirstPoI_Eucl	37	distFromFirstPoI_Lon	55	ratioPoIInUserPhotos	122
distFromFirstPoI_Lon	36	distFromFirstPoI_Eucl	54	stopProb	107

Results in Table IX show that the two features freqBigrams and freqTrigrams are the two most discriminative signals in the three cities examined. In particular, these features and the features distFromLastPoI are the same of those discovered in the *a priori* analysis based on Pearson's correlation. Not surprisingly, the analysis on GBRT-based models points out that the features freqBigrams and freqTrigrams are the most significative signals capturing popularity since they model both the global importance of a PoI and its relative importance w.r.t. those that have been already visited. In general, the model uses different signals regarding sequences of PoIs as well as distance, popularity and entropy indicators. Table X presents the top most recurrent features in each model and confirms once again the prevalence of the features previously discussed.

Tables IX and X highlight the different types of features that are in the top-10 positions per city. In particular, for Pisa and Florence (small and medium-size cities) distance-based features like, for example, distFromLastPoI_Eucl and distFromFirstPoI_Eucl are discriminative in modeling the behavior of a tourist. Surprisingly, this class of features is not as important for bigger cities (i.e., Rome). This observation suggests that the size of the city affects the way the tourist chooses to visit it. In small cities, where PoIs are close to each other, tourists take distances more into account (as they usually walk from a PoI to the next one) while in bigger city distances are less important. Here, the behavior of the "average" tourist is more popularity-driven.

We now investigate how our method behaves without employing the top 5 most significant features, we take as example Florence, and then adding each one step by step. To do so, we employ the top 5 features from Table IX that are: freqBigrams, freqTrigrams, entropy, visitTime_Max and ratioPoIInUserPhotos. These 5 features are found to be extremely relevant to the models due to their discriminative power.

In Table XI, we experiment with training models on datasets which do not contain either one or all of top 5 features. We see that the effectiveness of LEARNEXT drops significantly when we take out the top 5 most important features from the training set. Once we start adding each one of the top 5 features, we see results vary. The biggest contribution to the effectiveness is brought on by adding the feature which captures short sequences of 3 PoIs, freqTrigrams followed by the one capturing sequences of 2 PoIs, freqBigrams. By adding freqTrigrams to the set of features without the top 5 most important ones, the performance in terms of Success@1 triples. However, it is not enough to achieve the maximum effectiveness as when using all 68 features. On the contrary, the improvement brought on by entropy, visitTime_Max and ratioPoIInUserPhotos is rather low.

Table XI. Effectiveness (in terms of Success@k) of the models by varying the top 5 features for Florence used for modeling the tourist behavior.

Features	Success				
	@1	@2	@3	@5	@10
All - top 5	9.18%	13.45%	17.12%	21.63%	28.73%
All - top 5 + entropy	9.44%	13.78%	16.37%	21.05%	27.81%
All - top 5 + visitTime_Max	9.60%	13.70%	16.29%	21.38%	28.98%
All - top 5 + freqBigrams	14.87%	21.21%	26.56%	34.75%	47.28%
All - top 5 + freqTrigrams	34.50%	42.43%	46.95%	51.54%	57.14%
All - top 5 + ratioPoIInUserPhotos	9.85%	13.61%	17.20%	21.80%	29.65%

Pairwise mutual relationships between features. We introduce a novel technique that takes into account the Pearson's correlation between pairs of features. To assess the sensitivity of the model w.r.t. the features, we selectively eliminate a subset of features from the training set and we then learn and evaluate the performance.

We define an undirected graph where nodes are features and edges are weighted by the Pearson's correlation between pairs of features. Edges are then removed if their weight is below a given threshold ρ. In the resulting pruned graph we extract the connected components. From each connected component we choose one node as its representative. In particular, we choose as representative the node with highest Pearson's correlation w.r.t. the training label. Note that after pruning the graph some nodes could be disconnected. We use these resulting individual nodes together with the set of representatives because, even if they are not correlated with other features, they may withhold predictive power by themselves. We call this technique "Graph" and we test this approach on the city of Florence, chosen for its medium size and performance. We compare it with:

— **Random** - a simple baseline that chooses randomly the set of features to use in the model.
— **SelectKBest** - a univariate feature selection method which selects features according to the k highest scores, based on the Anova F-value statistical test.
— **Tree-Based** - a tree-based feature selection method which employs tree-based estimators and forests of trees in order to compute feature importance.
— **InfoGain** - a feature selection method, which employs an information gain loss function within forests of trees in order to compute feature importance.
— **GBFS** - Gradient Boosted Feature Selection, a state-of-the-art method for feature selection based on a modification of Gradient Boosted Trees [Xu et al. 2014]. Our tests employed the code kindly made available from the authors of the technique.

When comparing all the strategies, we fix the ρ parameter of our Graph method, which in turn determines the number of features selected (the first column of Tables XII, XIII). We then employ the same number of features for all the other feature selection strategies.

Results in Tables XII and XIII highlight the fact that all methods have a better performance than Random. The Graph method always outperforms the random baseline and the difference in terms of Success@k is significant.

Note that lower values of ρ means a deeper pruning of the feature space (i.e., a lower number of features to be used in the model). The resulting connected components have a higher number of nodes from which to choose the representative one. For $\rho = 0.17$, the method seems to be less accurate than all other feature selection methods, due to the fact that it build a long connected component, containing almost all features, from which extracts just one representative feature. If the top 5 feature belong to the same connected component then only one of them will be selected. However as we increase the degree of correlation between pairs the Graph method improves performance, the difference between Graph and all other methods not being statistically significant. The ability to select highly significant features is proven by the fact that by selecting 21 features the effectiveness of the prediction is even better than the one obtained by employing all the features, even though the difference is not statistically significant.

When computing feature selection, we used both GBFS with implicit parameter tuning as proposed by the authors and GBFS with a parameter setting similar to GBRT. When using the existing parameters we obtain 5 significant features and when we set out parameters we obtain 51 features from the feature selection process. Using all other feature selection method we select top 5 features and compare the results. The best performance is obtained however by the Tree-based feature selection method which reaches 33.75% in Success@1, when training with GBRT. Although using 5 features yields good results, they can be further improved for example by using 21 features, the difference best feature selection method for 5 features and the best for 21 features being of approximately 4%.

Regarding Ranking SVM (RSVM), the methods behave similarly in proportion as when training with GBRT. We have also computed the statistical significance for the values obtained with every feature selection method, for both RSVM and GBRT, and in most cases the difference is not statistically significant (p-value = 0.05), except when using 5 features and Graph has less in Success@k than the others: InfoGain, Tree-Based, SelectKBest and GBFS.

As the number of features increases, the performance is maintained around 37%-38% so there is no significant improvement with adding more features. By looking at the global MRR, values are very similar with the exception of Random and Graph with 5 features, for GBRT they vary by 0.01, while for RSVM they vary by maximum 0.02. MRR also has the same in the case of GBRT with Graph (ρ = 0.5), Graph (ρ = 0.6).

For top performance we can largely say that Graph, SelectKBest, Tree-Based, InfoGain and GBFS behave similarly. We observed the behavior of the proposed techniques on a various number of features and after computing the statistical significance of results we can say that results are "statistically" the same as when using 68 features.

In conclusion we answer **RQ5** by stating that *the method we propose for solving the* LEARNEXT *problem captures the overall behavior of a tourist. The performance of the model remains constant for a smaller number of features if they are "well-chosen". Similar performance can be obtained with less features by using feature selection methods as the ones presented above.*

Table XII. Effectiveness of the proposed feature selection techniques on the Florence dataset w.r.t random by applying GBRT. We report Success@k, MRR@k and total MRR.

# Feat.	Method		Success (MRR)					MRR
			@1	@2	@3	@5	@10	
5	GBRT	Graph ($\rho = 0.17$)	23.30% (0.23)	30.57% (0.26)	34.75% (0.28)	38.17% (0.29)	41.93% (0.29)	0.29
		Random	4.23% (0.04)	5.78% (0.04)	8.56% (0.06)	10.45% (0.07)	13.96% (0.07)	0.08
		SelectKBest	31.99% (0.31)	38.42% (0.35)	43.27% (0.36)	47.86% (0.37)	53.04% (0.38)	0.39
		Tree-Based	33.75% (0.33)	43.52% (0.38)	49.95% (0.40)	58.14% (0.42)	67.00% (0.43)	0.45
		InfoGain	32.99% (0.32)	43.10% (0.38)	49.12% (0.40)	57.56% (0.41)	66.49% (0.43)	0.44
		GBFS	30.24% (0.30)	38.17% (0.34)	42.02% (0.35)	46.11% (0.36)	51.62% (0.37)	0.38
21	GBRT	Graph ($\rho = 0.5$)	38.01% (0.38)	46.95% (0.42)	52.21% (0.44)	59.56% (0.45)	67.75% (0.47)	0.48
		Random	13.61% (0.13)	18.27% (0.15)	21.92% (0.17)	26.43% (0.18)	34.03% (0.19)	0.20
		SelectKBest	37.00% (0.37)	46.44% (0.41)	52.54% (0.43)	59.73% (0.45)	68.67% (0.46)	0.47
		Tree-Based	35.83% (0.35)	46.61% (0.41)	52.96% (0.43)	59.64% (0.44)	68.67% (0.46)	0.47
		InfoGain	34.67% (0.34)	45.27% (0.39)	52.63% (0.42)	58.73% (0.43)	68.58% (0.45)	0.46
27	GBRT	Graph ($\rho = 0.6$)	37.92% (0.37)	46.44% (0.42)	51.12% (0.43)	58.39% (0.45)	68.92% (0.46)	0.48
		Random	20.40% (0.20)	26.21% (0.23)	30.74% (0.24)	36.47% (0.26)	44.54% (0.27)	0.28
		SelectKBest	37.51% (0.37)	46.53% (0.42)	52.04% (0.43)	59.73% (0.45)	68.75% (0.46)	0.48
		Tree-Based	38.09% (0.38)	46.61% (0.42)	52.63% (0.44)	59.56% (0.45)	69.08% (0.47)	0.48
		InfoGain	35.00% (0.35)	44.36% (0.39)	51.46% (0.42)	58.81% (0.43)	69.34% (0.45)	0.46
39	GBRT	Graph ($\rho = 0.7$)	36.17% (0.36)	44.86% (0.40)	51.12% (0.42)	58.31% (0.44)	68.50% (0.45)	0.46
		Random	22.32% (0.22)	29.67% (0.25)	34.46% (0.27)	41.30% (0.29)	51.87% (0.30)	0.32
		SelectKBest	35.67% (0.35)	46.78% (0.41)	53.13% (0.43)	60.06% (0.44)	68.42% (0.46)	0.47
		Tree-Based	36.59% (0.36)	46.28% (0.41)	52.46% (0.43)	58.89% (0.44)	69.08% (0.46)	047
		InfoGain	36.50% (0.36)	46.36% (0.41)	52.46% (0.43)	60.73% (0.45)	69.42% (0.46)	0.47
47	GBRT	Graph ($\rho = 0.8$)	37.34% (0.37)	45.19% (0.41)	50.37% (0.42)	58.81% (0.44)	68.92% (0.46)	0.47
		Random	28.07% (0.28)	36.15% (0.32)	41.75% (0.33)	49.10% (0.35)	59.73% (0.37)	0.38
		SelectKBest	35.67% (0.35)	45.11% (0.40)	51.62% (0.42)	58.73% (0.44)	68.58% (0.45)	0.46
		Tree-Based	36.59% (0.36)	46.36% (0.41)	52.21% (0.43)	59.23% (0.45)	68.58% (0.46)	0.47
		InfoGain	36.84% (0.36)	46.36% (0.41)	52.46% (0.43)	60.15% (0.45)	69.17% (0.46)	0.47
51	GBRT	Graph ($\rho = 0.825$)	36.75% (0.36)	46.70% (0.41)	52.71% (0.43)	58.81% (0.45)	68.25% (0.46)	0.47
		Random	27.98% (0.27)	35.95% (0.31)	41.56% (0.32)	48.83% (0.35)	58.06% (0.36)	0.37
		SelectKBest	36.50% (0.36)	45.69% (0.41)	53.46% (0.43)	59.98% (0.45)	69.84% (0.46)	0.47
		Tree-Based	36.59% (0.36)	46.03% (0.41)	52.54% (0.43)	60.23% (0.45)	68.75% (0.46)	0.47
		InfoGain	37.42% (0.37)	47.11% (0.42)	52.96% (0.44)	59.64% (0.44)	69.25% (0.47)	0.48
		GBFS	35.42% (0.35)	45.44% (0.40)	52.96% (0.42)	60.06% (0.44)	69.59% (0.45)	0.46
57	GBRT	Graph ($\rho = 0.9$)	36.92% (0.36)	46.28% (0.41)	51.87% (0.43)	58.56% (0.44)	68.33% (0.46)	0.47
		Random	27.80% (0.27)	36.14% (0.31)	42.23% (0.34)	49.64% (0.35)	60.18% (0.37)	0.38
		SelectKBest	37.17% (0.37)	45.69% (0.41)	52.96% (0.43)	59.98% (0.45)	70.59% (0.46)	0.47
		Tree-Based	37.76% (0.37)	47.53% (0.42)	53.04% (0.44)	60.15% (0.46)	69.00% (0.47)	0.48
		InfoGain	37.17% (0.37)	46.11% (0.41)	52.38% (0.43)	59.56% (0.45)	69.17% (0.46)	0.47
68	GBRT	None	37.76% (0.37)	46.78% (0.42)	53.04% (0.44)	59.31% (0.45)	69.34% (0.47)	0.48

Table XIII. Effectiveness of the proposed feature selection techniques on the Florence dataset w.r.t. random by applying RSVM. We report Success@k, MRR@k and total MRR.

# Feat.		Method	Success (MRR)					MRR
			@1	@2	@3	@5	@10	
5	RSVM	Graph (ρ = 0.17)	23.72% (0.23)	31.16% (0.27)	34.92% (0.28)	38.93% (0.29)	42.43% (0.30)	0.30
		Random	3.45% (0.03)	4.80% (0.03)	5.65% (0.04)	7.45% (0.04)	9.84% (0.05)	0.05
		SelectKBest	29.82% (0.29)	36.84% (0.33)	39.34% (0.34)	44.27% (0.35)	48.95% (0.35)	0.36
		Tree-Based	32.49% (0.32)	43.44% (0.37)	48.12% (0.39)	54.21% (0.40)	61.65% (0.41)	0.43
		InfoGain	32.99% (0.32)	42.18% (0.37)	47.03% (0.39)	51.71% (0.40)	58.73% (0.41)	0.42
		GBFS	30.32% (0.30)	36.09% (0.33)	39.51% (0.34)	42.10% (0.34)	47.11% (0.35)	0.36
21	RSVM	Graph (ρ = 0.5)	34.92% (0.34)	44.36% (0.39)	49.20% (0.41)	54.38% (0.42)	60.81% (0.43)	0.44
		Random	9.90% (0.09)	13.91% (0.11)	16.39% (0.12)	21.21% (0.13)	27.95% (0.14)	0.16
		SelectKBest	33.75% (0.33)	41.35% (0.37)	45.19% (0.38)	50.87% (0.40)	57.39% (0.41)	0.42
		Tree-Based	34.25% (0.34)	42.27% (0.38)	45.69% (0.39)	50.71% (0.40)	56.97% (0.41)	0.42
		InfoGain	32.35% (0.32)	41.01% (0.36)	45.27% (0.38)	49.53% (0.39)	56.32% (0.39)	0.41
27	RSVM	Graph (ρ = 0.6)	35.17% (0.35)	44.52% (0.39)	49.12% (0.41)	53.88% (0.42)	59.23% (0.43)	0.44
		Random	17.27% (0.17)	22.90% (0.20)	25.61% (0.20)	30.55% (0.22)	38.06% (0.23)	0.24
		SelectKBest	34.58% (0.34)	42.52% (0.38)	46.11% (0.39)	50.54% (0.40)	56.55% (0.41)	0.42
		Tree-Based	34.25% (0.34)	42.35% (0.38)	45.44% (0.39)	49.54% (0.40)	56.14% (0.41)	0.42
		InfoGain	33.39% (0.33)	41.22% (0.37)	45.36% (0.38)	49.85% (0.39)	56.30% (0.40)	0.41
39	RSVM	Graph (ρ = 0.7)	34.50% (0.34)	44.61% (0.39)	49.28% (0.41)	54.05% (0.42)	59.89% (0.43)	0.43
		Random	19.83% (0.19)	25.49% (0.22)	28.92% (0.23)	33.08% (0.24)	41.25% (0.25)	0.27
		SelectKBest	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42
		Tree-Based	34.33% (0.34)	41.93% (0.38)	45.02% (0.39)	49.03% (0.40)	55.97% (0.41)	0.42
		InfoGain	34.00% (0.34)	41.77% (0.37)	45.61% (0.39)	50.12% (0.40)	56.47% (0.41)	0.42
47	RSVM	Graph (ρ = 0.8)	35.08% (0.35)	43.52% (0.39)	47.03% (0.40)	52.13% (0.41)	58.06% (0.42)	0.43
		Random	24.85% (0.24)	31.86% (0.28)	35.52% (0.29)	40.11% (0.30)	47.26% (0.31)	0.32
		SelectKBest	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42
		Tree-Based	34.33% (0.34)	41.93% (0.38)	45.02% (0.39)	49.03% (0.40)	55.97% (0.41)	0.42
		InfoGain	34.08% (0.34)	42.02% (0.38)	45.53% (0.39)	49.87% (0.40)	56.05% (0.40)	0.42
51	RSVM	Graph (ρ = 0.825)	35.08% (0.35)	43.52% (0.39)	47.03% (0.40)	52.13% (0.41)	58.06% (0.42)	0.43
		Random	24.36% (0.24)	31.22% (0.28)	36.04% (0.29)	40.67% (0.30)	47.96% (0.31)	0.32
		SelectKBest	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42
		Tree-Based	34.25% (0.34)	42.77% (0.38)	45.69% (0.39)	50.45% (0.40)	56.89% (0.41)	0.42
		InfoGain	34.58% (0.34)	42.43% (0.39)	46.19% (0.39)	50.37% (0.40)	55.80% (0.41)	0.42
		GBFS	33.39% (0.33)	41.22% (0.37)	45.36% (0.38)	49.85% (0.39)	56.30% (0.40)	0.41
57	RSVM	Graph (ρ = 0.9)	34.41% (0.34)	42.43% (0.38)	46.03% (0.39)	50.71% (0.40)	58.31% (0.41)	0.42
		Random	24.71% (0.24)	31.54% (0.28)	34.87% (0.29)	39.29% (0.30)	46.59% (0.31)	0.32
		SelectKBest	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42
		Tree-Based	34.25% (0.34)	42.77% (0.38)	45.69% (0.39)	50.45% (0.40)	56.89% (0.41)	0.42
		InfoGain	34.58% (0.34)	42.43% (0.38)	46.19% (0.39)	50.37% (0.40)	55.80% (0.41)	0.42
68	RSVM	None	34.58% (0.34)	42.02% (0.38)	45.69% (0.39)	49.37% (0.40)	56.47% (0.41)	0.42

5. CONCLUSIONS AND FUTURE WORK

We proposed to apply machine learning techniques to tackle the problem of predicting the next tourist attraction a tourist will visit on the basis of her visit history (i.e., the prediction is done accordingly to what a tourist has already visited). This application could be of help in different scenarios, e.g., i) prediction of tourist flows, ii) location advertising. In the first scenario, our predictor can be used to devise how tourists will visit the city from a macroscopic point of view thus helping the management of the resources of the city dedicated to tourism, while the second scenario relies on exploiting a tourist prediction for understanding the effect of the advertisement in a particular part of the city or, more importantly, for choosing where to place it in order to maximize its effects.

We modeled the task as an instance of *learning to rank* problem and we defined a feature space composed of 68 features capturing both the tourist behavior and the peculiar characteristics of candidate PoIs. In particular, we built the ranking models by exploiting two well-know techniques: Ranking SVM [Joachims 2006] and Gradient Boosted Regression Trees (GBRT) [Zheng et al. 2007]. GBRT and Ranking SVM consistently outperform the PROB baseline by 300% in terms of prediction accuracy. Our methods also outperform the proposed competitors: “WhereNext”, “Random Walk”, “Logistic Regression” and “SVM-C”. We tested the effectiveness of the proposed solutions on three important tourist cities (Pisa, Florence, and Rome) and we observed a consistent gain in performance compared to all other methods.

Moreover to analyzing the performance in comparison with the baselines, we test the robustness of our approach by answering five different research questions.

— The first RQ is related to the efficiency of learning-to-rank models for predicting the next PoI. In order to answer these question we compare our propose methods with a probability baseline and four competitors. Our methods always outperform our competitors thus we can conclude that learning to rank methods are efficient for predicting the “next” PoI of a user visiting a city.
— The second RQ regards the tuning of the parameters for each learning algorithm and their impact on the performance. We observe that although they have a certain influence, the performance results, especially in the case of GBRT for the small and medium dataset, are not significantly different. For Ranking SVM we also see that results tend to stabilize, so once achieved the minimum threshold, performance varies slightly, but the difference is not statistically significant. Nevertheless, a good tuning of parameters in accordance with the size of the dataset is necessary for attaining good results.
— The next RQ takes into account how results are affected by the length of the trails. According to our tests results improve with more historical information about the current trails, the best performance is achieved after the tourist has already visited 5-7 PoIs in respect with just one. However, a reasonably good prediction can be made with having only 3 PoIs in the current trail.
— Another RQ makes the distinction between predicting rare or popular PoIs. As one would expect, it is easier to predict more popular PoIs than rare PoIs, but our system succeeds in coping with rare PoIs as well, and offer a performance of more than 30% for the small and medium dataset and more that 20% for the large dataset.
— The last RQ makes a thorough analysis of the sensitivity of the models with respect to the features. We analyze features prior and after training, and determine the ones which have a higher impact on each of the three datasets. We also propose a method for selecting the most discriminant features, which allows us to achieve similar performance when using only 21 features instead of 68, and compare it with other feature selection methods and a random baseline. We conclude that our fea-

tures succeed in covering several aspects involved in a tourist visit and the overall behaviour of a tourist, but at the same time, by using a method for feature selection, thus reducing the dimensionality of the feature space, we can attain comparable performances.

We are investigating possible applications where our proposed technique can be a fruitful building block. A promising use of this research is to devise a method to plan a visit in a city ahead of time by using LEARNEXT as a building block.

REFERENCES

Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Hui-I Yang. 2007. World explorer: visualizing aggregate data from unstructured text in geo-referenced collections. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 1–10.

Yuki Arase, Xing Xie, Takahiro Hara, and Shojiro Nishio. 2010. Mining People's Trips from Large Scale Geo-tagged Photos. In *Proceedings of the International Conference on Multimedia (MM '10)*. ACM, New York, NY, USA, 133–142. DOI:http://dx.doi.org/10.1145/1873951.1873971

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, and others. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

Ranieri Baraglia, Claudio Frattari, Cristina Ioana Muntean, Franco Maria Nardini, and Fabrizio Silvestri. 2012. A Trajectory-Based Recommender System for Tourism. In *Proc. AMT 2012*.

Ranieri Baraglia, Cristina Ioana Muntean, Franco Maria Nardini, and Fabrizio Silvestri. 2013. LearnNext: Learning to Predict Tourist Movements. In *Proc. CIKM*. ACM, 6.

Christopher M Bishop and others. 2006. *Pattern recognition and machine learning*. Vol. 1. springer New York.

Igo Ramalho Brilhante, José Antônio Fernandes de Macêdo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2013. Where shall we go today?: planning touristic tours with tripbuilder. In *CIKM*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 757–762.

Jonathan Chang and Eric Sun. 2011. Location 3: How users share and respond to location-based data on social networking sites. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*. 74–80.

Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. *JMLR* 14 (2011), 1–24.

Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1082–1090.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. 2009. Mapping the world's photos. In *Proc. WWW*. ACM, 761–770.

Huiji Gao, Jiliang Tang, and Huan Liu. 2012. Exploring Social-Historical Ties on Location-Based Social Networks.. In *ICWSM*.

Fabien Girardin. 2009. *Aspects of implicit and explicit human interactions with ubiquitous geographic information*. Ph.D. Dissertation.

Fabien Girardin, Francesco Calabrese, Filippo Dal Fiore, Carlo Ratti, and Jose Blat. 2008a. Digital footprinting: Uncovering tourists with user-generated content. *Pervasive Computing, IEEE* 7, 4 (2008), 36–43.

Fabien Girardin, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. 2008b. Leveraging explicitly disclosed location information to understand tourist dynamics: a case study. *Journal of Location Based Services* 2, 1 (2008), 41–56.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. SIGKDD*. ACM, 133–142.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proc SIGKDD (KDD '06)*. ACM, New York, NY, USA, 217–226.

John Krumm and Eric Horvitz. 2006. Predestination: inferring destinations from partial trajectories. In *Proc. UbiComp'06*. Springer-Verlag.

Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. 2012. Lars: A location-aware recommender system. In *Proc. ICDE*. IEEE, 450–461.

Defu Lian, Xing Xie, Vincent W. Zheng, Nicholas Jing Yuan, Fuzheng Zhang, and Enhong Chen. 2015. CEPR: A Collaborative Exploration and Periodically Returning Model for Location Prediction. *ACM Trans. Intell. Syst. Technol.* 6, 1, Article 8 (April 2015), 27 pages. DOI:http://dx.doi.org/10.1145/2629557

Defu Lian, Yin Zhu, Xing Xie, and Enhong Chen. 2014. Analyzing Location Predictability on Location-Based Social Networks. In *Advances in Knowledge Discovery and Data Mining*. Springer, 102–113.

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225–331. DOI:http://dx.doi.org/10.1561/1500000016

Claudio Lucchese, Raffaele Perego, Fabrizio. Silvestri, Hossein. Vahabi, and Rossano Venturini. 2012. How Random Walks can Help Tourism. In *Proc. ECIR*. LNCS.

Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. WhereNext: a location predictor on trajectory pattern mining. In *Proc. SIGKDD*. ACM.

Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. Mining User Mobility Features for Next Place Prediction in Location-Based Services. In *Proc. ICDM*. IEEE Computer Society.

Feng Pan, Tim Converse, David Ahn, Franco Salvetti, and Gianluca Donato. 2009. Feature selection for ranking using boosted trees. In *Proc. CIKM*. ACM, 4.

Marco Pennacchiotti, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. 2012. Making your interests follow you on twitter. In *Proc. CIKM*. ACM, New York, NY, USA, 165–174.

Adrian Popescu and Gregory Grefenstette. 2009. Deducing trip related information from flickr. In *Proc. WWW*. ACM.

Tye Rattenbury, Nathaniel Good, and Mor Naaman. 2007. Towards automatic extraction of event and place semantics from flickr tags. In *Proc. SIGIR*. ACM.

Adam Sadilek, Henry Kautz, and Jeffrey P Bigham. 2012. Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 723–732.

Zhixiang Xu, Gao Huang, Kilian Q. Weinberger, and Alice X. Zheng. 2014. Gradient Boosted Feature Selection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 522–531. DOI:http://dx.doi.org/10.1145/2623330.2623635

Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proc. SIGSPATIAL*. ACM.

Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc. SIGIR*. ACM.

Yu Zheng, Xing Xie, and Ma Wei-Ying. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. WWW*. ACM, 791–800.

Yu Zheng and Xie Xing. 2011. Learning travel recommendations from user-generated GPS traces. *ACM TIST* 2, 1 (2011), 2.

Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier. Chapelle, Keke Chen, and Gordon Sun. 2007. A general boosting method and its application to learning ranking functions for web search. *Advances in neural information processing systems* 20 (2007), 1697–1704.

Received X; revised X; accepted X